Adaptive Multigroup Radiation Diffusion

by

Richard B. Williams

Submitted to the Department of Nuclear Science and Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Science in Nuclear Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Author .....
Department of Nuclear Science and Engineering
April 29, 2005

Certified by .....
Kim Molvig
Associate Professor of Nuclear Science and Engineering
Thesis Supervisor

Read by .....
Sidney Yip
Professor of Nuclear Science and Engineering
Thesis Reader

Accepted by .....
Jeffrey Coderre
Chairman, Department Committee on Graduate Students

Adaptive Multigroup Radiation Diffusion

by

Richard B. Williams

Submitted to the Department of Nuclear Science and Engineering
on April 29, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Science in Nuclear Science and Engineering

Abstract

This thesis describes the development and implementation of an algorithm for dramatically increasing the accuracy and reliability of multigroup radiation diffusion simulations at low group counts. This is achieved by allowing the energy group boundaries to move in energy space as the simulation evolves. This adaption in energy space effectively removes the sensitivity of multigroup diffusion to group boundary placement and makes the technique a viable option for large, computationally expensive computer simulations.

Traditional multigroup radiation diffusion solvers break down at small group counts because of the coarse discretization of highly nonlinear material opacity. Small changes in the group boundary energies can lead to wildly different mean opacities and therefore significant changes in simulation output. This sensitivity has rendered the technique unpredictable and it is generally considered to be not worth the added computational expense.

Unfortunately, multigroup diffusion is the only method available for adding frequency dependence to the radiation field in a standard diffusion solver. When attempting to model a system that includes a non-equilibrium radiation field, frequency-dependent effects become important. While running multigroup simulations with large numbers of groups is a reliable method for obtaining increased accuracy over grey diffusion, the computational expense scales linearly with the number of groups. For large simulations, running multigroup diffusion with increased group counts is infeasible and running with small group counts is unreliable. This has led to a tendency to use grey diffusion even in environments where the radiation field is known to be out of equilibrium with the material.

This thesis includes a new derivation of the diffusion equation and an overview of traditional "static" multigroup radiation diffusion along with an analysis of its shortcomings. The sensitivity due to group boundary placement for small numbers of groups is shown. Data are presented which demonstrate that small group count multigroup calculations can actually provide a worse answer than grey diffusion.

A system is developed and implemented for allowing the multigroup energy boundaries to "adapt," or move in energy space, as the simulation evolves as well as a

method for determining where increased energy resolution is needed for an arbitrary set of material opacities. By adapting in energy space, the sensitivity of multigroup diffusion to group boundary placement is ameliorated.

Data are presented that demonstrate a reliable increase in accuracy for adaptive multigroup diffusion as the number of groups is increased—even at very small group counts. Furthermore, the data show that the level of accuracy obtained with the adaptive multigroup approach is equivalent to or better than the best-case data obtained with the static multigroup approach.

This result is more profound than a simple increase in accuracy—the increased reliability makes multigroup radiation diffusion a viable tool for large non-equilibrium simulations. Users are no longer forced to use the grey diffusion method which is known to be physically inadequate. Furthermore, users are finally free to incorporate a frequency-dependent treatment of radiation without the accuracy ambiguities of standard multigroup diffusion.

Thesis Supervisor: Kim Molvig
Title: Associate Professor of Nuclear Science and Engineering

Acknowledgments

Prior to my change of station to New Mexico, I was ensconced in the standard graduate routine at MIT. The bulk of my classmates had nuclear engineering undergraduate degrees and, as such, I was forced to do a certain amount of catch-up in order for me to pass my qualifying exams in the fall of 2001. I could not have succeeded at this without the patient tutoring of my roommate, Evan Fortunato. Professor Dick Lanza was a needed source of support during those hectic weeks leading up to my oral examination. I'd like to thank Professors Sidney Yip and Jeff Freidberg for agreeing to sit on my oral exam committee and for their helpful commentary on the future of my graduate career. No one can achieve anything in the Nuclear Science and Engineering Department without the help of Clare Egan, who's logistical wizardry was essential for me to navigate the sometimes byzantine MIT bureaucracy. Finally, I'd like to thank Jack Florey and the denizens of Cruftlabs for keeping me inspired and sane during my time in Cambridge.

When I moved to Los Alamos in the spring of 2002 to pursue my thesis research full time, I found myself surrounded by an amazing cadre of scientists. Having the opportunity to discuss my theories with the likes of Richard Bowers, Gordon Olson, Dimitri Mihalas, and Jim Morel was incredible, and I thank them immensely for their advice. My first mentor at the laboratory, John Hall, deserves a tremendous amount of thanks for helping me get settled in the LANL system and supporting me during my first two summers here. Once I transferred into the Thermonuclear Applications group, I had the pleasure of working under Brad Beck and Dave Harris, the most effective management team I have ever experienced. I'd also like to thank my mentor in X-2, Bob Weaver, for helping me successfully navigate the sometimes unforgiving political atmosphere of X Division. I'd like to thank my office neighbors Chriss Bernardin and Rob Coker, each of whom saved me from almost certain insanity at least once. Chris Fontes, whom I met the first day I came to Los Alamos, went from being my ultimate frisbee instructor to the only LANL staff member on my thesis committee. He is responsible for my understanding of atomic physics and material

opacity, as well as my ability to throw the high-release forehand. In general, I need to thank Los Alamos National Laboratory[1] for the use of their facilities and financial support of my research.

Not to be left out is the assistance I received from the folks at SAIC in San Diego. Tom Betlach, who is responsible for the existing multigroup radiation diffusion solver in RAGE, was an endless resource and I thank him for putting up with my infinite series of questions about the code. Mike Clover and Nelson Byrne also deserve thanks for their participation in brainstorming sessions about my theory. Of course no mention of SAIC's involvement in RAGE is complete without Mike Gittings, the mastermind behind it all. Mike's knowledge of the infrastructure of the RAGE code framework was the only thing that allowed me to integrate my algorithm and I can't thank him enough for his help and guidance.

Eventually, I did manage to put together something I'm proud of and a special thanks goes out to my thesis committee, who helped shape the document and the research as a whole. Their input was essential in my success as a graduate student.

During the course of my three years living in New Mexico, there were a few people who made my life interesting outside of the academic realm and I wish to thank them for forcing me to occasionally leave work. Christina Scovel and Robin Blume-Kohout are two of the most unusual people I have ever met. Christina fed me chocolate when I needed it and helped get me my postdoc. Robin fed me steak and convinced me that I was capable of climbing mountains. Dan Driscoll and Lukas Bradley were always there to offer needed distraction, insight, and support. Finally, I owe a tremendous amount to Nina, whose love and companionship over the last three years has made all the difference in the world. I could not succeed without her support.

[1]This document is cleared for release by LANL and has been assigned the unlimited release number LA-UR-05-2884.

Contents

1 Introduction **15**

2 Properties of the Radiation Field **17**

2.1 Photon Number Density and Specific Intensity 17

2.2 Radiation Energy Density 18

2.3 Radiative Energy Flux 19

2.4 Radiation Pressure Tensor 20

3 The Equation of Radiation Transfer **23**

3.1 Absorption, Scattering, and Total Opacities 23

3.2 Emission 25

3.3 The Transfer Equation 26

3.4 Moments of the Transfer Equation 27

3.5 Local Thermal Equilibrium 28

3.6 The Diffusion Limit 30

3.7 A contrasting derivation of the diffusion equation 51

4 Flux Limited Diffusion **55**

4.1 The Variable Eddington Factor 56

4.2 Flux Limiting 58

5 Grey Transport **61**

5.1 Opacity Spectrum Complexity 62

5.2 Mean Opacities 65

6 Multigroup Transport **69**

6.1 Group Boundary Placement 73

6.2 Equal Arc Length Projection 74

6.3 Dealing with changes in the opacity spectrum 76

7 Adaption in Energy Space **79**

7.1 Accounting for spatial opacity variation 79

7.2 Accounting for temporal opacity variation 81

7.3 Mechanics of altering group structure 82

7.3.1 Evaluation of discrete opacity data 82

7.3.2 Discrete equal arc-length projection 83

7.3.3 Generating the coarse opacity table 85

7.3.4 Handling group-dependent variables 88

8 Implementing Adaptive Multigroup Radiation Diffusion in RAGE **91**

8.1 Multigroup Radiation Diffusion in RAGE 92

8.2 Opacities in RAGE and the TOPS code 95

8.3 Evaluating the overhead of the Adaptive Multigroup method 96

8.3.1 Order of magnitude computational cost: Adaptive Multigroup 97

8.3.2 Order of magnitude computational cost: Multigroup Diffusion 100

8.4 Splicing the Adaptive Multigroup module into RAGE 101

9 Performance Comparisons **103**

9.1 A Figure of Merit 103

9.2 A Simple Test Problem: The Marshak Wave 104

9.2.1 Multigroup Solution 106

9.3 Static Multigroup and the Marshak Wave 107

9.3.1 Manually choosing photon energy group boundaries 121

9.3.2 Sensitivity of results to group boundary placement 122

9.4 Adaptive Multigroup and the Marshak Wave 125

9.4.1 Parameter study for adaptive multigroup 126

9.4.2 Varying the number of coarse groups 126
9.4.3 Varying the number of high-resolution groups 128
9.4.4 Adaptive vs. Static Results 130

10 Conclusion **135**

10.1 Future Investigations 135
10.1.1 Performance Upgrades 135
10.1.2 Full RAGE integration 136
10.1.3 Accuracy improvements 137
10.2 Summary 138

A Source Code for Adaptive Multigroup Radiation Diffusion in RAGE 141

A.1 Module Initialization 141
A.2 The Regroup Routine 148
A.3 The Coarsen Routine 161
A.4 Initialization of the Radiation Energy Density Master Table 177
A.5 Coarsening of the Radiation Energy Density 180
A.6 Refining the changes in Radiation Energy Density 183

List of Figures

5-1 Iron opacity for $T = 0.1$ keV and $\rho = 0.0004$ g/cm^3. The opacities in these plots have been normalized by the material density, hence the change in units from the 1/cm seen previously. 62

5-2 Iron opacity for $T = 5$ keV and $\rho = 0.0004$ g/cm^3 64

5-3 Iron opacity for $T = 0.5$ eV and $\rho = 0.0004$ g/cm^3 65

6-1 A small section of an opacity plot (solid line) superimposed with a corresponding set of multigroup opacities (dashed line). Notice that in this example, the group boundaries are uniformly spaced in log-energy space. Notice also that the discontinuous function represented by the multigroup opacities maintains the general character of the original opacity function, but misses smaller features such as the spike just below 1keV. 72

6-2 An example opacity on which the equal arc length projection method has been performed. The vertical lines indicate the group boundary selections made by the algorithm. Notice that the region of dense bound-bound transition lines has very high group resolution while the relatively smooth regions are characterized by broad energy groups. . 76

9-1 The radiation energy density as a function of position. In this example, the driving wall temperature is 100 eV. 108

9-2 The integrated radiation energy density as a function of position for multigroup transport. In this example, the driving wall temperature is 100 eV. 109

9-3 The multigroup radiation energy density in group 9, spanning photon energies from 100 keV to 316.2 keV. Note the exponent on the ordinate values—this is at the limit of machine precision and consists entirely of round-off artifacts. 110

9-4 The multigroup radiation energy density in group 1, spanning photon energies from 0.1 eV to 31.62 eV. 111

9-5 The material temperature at the time when the integrated radiation energy density propagation front reaches 0.5 cm. 112

9-6 The frequency-integrated radiation energy density for a Marshak wave with a 500 eV driving wall temperature, showing the difference in output for grey and 10-group multigroup diffusion simulations. The time required for the grey solution to propagate half-way through the slab, as shown in this plot, was 2.4×10^{-11} s. 113

9-7 The multigroup radiation energy densities for a 10-group Marshak wave with a 500 eV driving wall temperature, showing each group's contribution to the integrated total. The numbers in parenthesis next to each group label are the group boundaries for that group. All ten groups are plotted on the same scale; those not listed in the legend contributed too little to appear. 115

9-8 The Planck spectrum for a 500 eV wall temperature. The vertical lines are the group boundaries used in the initial 500 eV Marshak wave simulation. 116

9-9 The frequency-integrated radiation energy density for two ten-group simulations with different high-energy limits. The solid line data has a high energy limit of 1000 keV. The dashed line has a high energy limit of 30 keV. Both high energy limits are in a region of energy space which contains negligible photon populations. 118

9-10 The frequency-integrated radiation energy density for two ten-group simulations with different high-energy limits is compared to a high-resolution 1000-group simulation. The solid line data is the 10-group run with an energy limit of 70 keV. The dashed line is the 10-group run with an energy limit of 30 keV. The dotted line is the 1000-group high resolution data. 119

9-11 The opacity for iron at a density of 0.005 g/cm^3 and two different temperatures. These temperatures represent the starting "cold" slab and the equilibrium temperature of the hot driving wall. The two opacity data sets have different limits as is shown here. The TOPS code extrapolates along a curve of ν^{-3} for opacities off the table. . . . 121

9-12 The normalized RMS error in radiation energy density as a function of group count. 123

9-13 The normalized RMS error in radiation energy density as a function of group count. Comparison of two different energy limits at small group counts. 124

9-14 An examination of the falloff in accuracy as the number of coarse groups is decreased. The number of high-resolution groups and the energy limits are held constant. 127

9-15 An examination of the improvement in accuracy available by enabling adaptive multigroup for the 500 eV Marshak wave. These simulations were conducted at the best-case energy limits for the 10-group static multigroup data. 130

9-16 An examination of the improvement in accuracy available by enabling adaptive multigroup for the 500 eV Marshak wave. These simulations were conducted at the worst-case energy limits for the 10-group static multigroup data. 132

9-17 Two static multigroup diffusion data sets compared against an adaptive multigroup diffusion data set. The best accuracy points on both static plots still have higher error than the corresponding adaptive multigroup data points. 133

Chapter 1

Introduction

Accurate radiation[1] transport simulations play a crucial role in high energy density physics. This includes areas of research such as astrophysics[36] and laser physics as well as emerging technologies like inertial confinement fusion[2, 18, 19]. In each of these examples, the material energy density is strongly coupled to the radiation field. In fact, the interaction between radiation and the material is so strong that the radiation tends to drive the hydrodynamics.

In order to understand the dynamics of such a high energy system, an accurate depiction of the radiation transport involved is critical. Furthermore, high energy density physics experiments tend to be prohibitively expensive (or, in the case of astrophysics, impossible). This expense generally leads to very few experimental data points being available and the bulk of the scientific endeavor must be done in simulation.

Unfortunately, radiation transport in current continuum radiation hydrodynamics simulations is handled in an overly-simplified manner which drastically impairs the predictive capability of many simulations. This is primarily a result of the enormous computational expense of high-accuracy radiation transport methods. While simplifying assumptions like the grey diffusion approximation make the computational issues more tractable, they are only valid in certain physical regimes. The multigroup

[1]Throughout this thesis, we adopt the vernacular of the computational physics community at Los Alamos National Laboratory, for whom "radiation" refers specifically to photons and does not include charged particles or neutrons.

diffusion method, which acts as a compromise between full transport and grey diffusion, does not return sufficient improvement in accuracy for low group counts to be worth the added computational cost. This thesis shows, for the first time, that when the number of energy groups is small, traditional static multigroup radiation diffusion's accuracy is not predictable and that unintuitive situations arise in which adding groups can lead to a *significant reduction* in the accuracy of the simulation.

This thesis develops a unique method for maximizing the accuracy gained from the multigroup radiation diffusion technique and removing the uncertainty in accuracy caused by a static selection of group boundaries. In simulation environments where the energy group count is limited, the accuracy gained is a function of the correlation between the opacity structure in the problem and the location of the energy group boundaries. If the problem being simulated involves significant changes in the opacity, then the accuracy gained by using multigroup diffusion can only be maintained if the group boundaries shift to accommodate the new opacity structure. The adaptive multigroup radiation diffusion code developed for this thesis is the first implementation of this concept. The static energy group boundaries used by a conventional multigroup diffusion package are modified dynamically to suit the current state of the simulation.

It is shown that this technique delivers improved accuracy when compared to standard multigroup diffusion with static group boundaries. This is equivalent to being able to use fewer groups to maintain a given level of accuracy, and thus reduces the computational cost of multigroup radiation diffusion. This increases the likelihood of higher-accuracy radiation transport approximations being used in lieu of grey diffusion. The sensitivity of the static multigroup method to group boundary placement when the group count is low has made many computational physicists reticent to use the technique despite it's capacity for a more accurate simulation of radiation transport. The introduction of adaption in energy space removes this sensitivity and thus provides the user with a predictable increase in accuracy for the increased computational cost of using multigroup diffusion.

Chapter 2

Properties of the Radiation Field

The interactions of photons and matter can result in significant changes to the local environment of the material. Even in simulations for which the material is considered to be static, the coupling between material and radiation can strongly influence the outcome. However, before we can consider the coupling of radiation and matter, we must first develop a description of the radiation field itself.

2.1 Photon Number Density and Specific Intensity

The radiation field is a complicated entity and is a function of position, time, propagation angle, and energy[24]. One way to look at the radiation field is as a collection of photons—each of which, at every instant, has a specific position, direction of propagation, and frequency. The quantity used to describe the radiation field in this way is the photon number density, $\mathcal{N}$. The photon number density is defined such that the number of photons per unit volume at location $\vec{x}$ and time t, with frequencies between ν and $\nu + d\nu$, propagating at velocity c in direction $\vec{n}$ into a solid angle $d\Omega$ is $\mathcal{N}(\vec{x}, t, \vec{n}, \nu)d\Omega d\nu$. The units of photon number density are $\mathrm{cm}^{-3}\ \mathrm{sr}^{-1}\ \mathrm{Hz}^{-1}$[22].

Each of the photons in the radiation field carries with it some energy, $E = h\nu$. If we take the photon number density and multiply it by $h\nu$, we produce the *radiation energy density*, $\mathcal{E}$. This is a measure of the energy contained in the radiation field per unit volume by photons at position $\vec{x}$, time t, moving in direction $\vec{n}$, and with

frequency ν. The relationship between photon number density and radiation energy density is simple but important:

$$\mathcal{E}(\vec{x}, t, \vec{n}, \nu) = h\nu\mathcal{N}(\vec{x}, t, \vec{n}, \nu) \tag{2.1}$$

If we take the radiation energy density and multiply it by the speed of light, c, we produce a quantity called the *specific intensity*, I.

$$I(\vec{x}, t, \vec{n}, \nu) = c\mathcal{E}(\vec{x}, t, \vec{n}, \nu) \tag{2.2}$$

The specific intensity is a measure of the transport of energy within the radiation field. It is defined such that the amount of energy transported by photons of frequencies between ν and $\nu + d\nu$ through a surface element dS into a solid angle $d\Omega$ in a time interval dt is

$$dE = I(\vec{x}, t, \vec{n}, \nu) dS cos\theta d\Omega d\nu dt \tag{2.3}$$

where θ is the angle between the propagation vector, $\vec{n}$, and the normal to the surface element, dS [22]. Specific intensity has units of ergs s^{-1} cm^{-2} Hz^{-1} sr^{-1}. As defined, the specific intensity gives a complete instantaneous description of the radiation field[1]. It is useful to examine various angular averages or *moments* of specific intensity which have physical significance.

2.2 Radiation Energy Density

Rearranging equation 2.2 gives us the following formula for the radiation energy density

$$\mathcal{E}(\vec{x}, t, \vec{n}, \nu) = \frac{I(\vec{x}, t, \vec{n}, \nu)}{c} \tag{2.4}$$

[1]This is not absolutely true as we have clearly ignored polarization in this definition. However, polarization is generally not considered in most aspects of computational radiation transport as its effects on radiation/matter coupling are usually negligible.

If we integrate this over all solid angles, we get

$$\mathcal{E}(\vec{x},t,\nu) = \frac{1}{c}\oint I(\vec{x},t,\vec{n},\nu)d\Omega \tag{2.5}$$

This is referred to as the *monochromatic radiation energy density* because it is still a function of photon frequency though the angular dependence is gone. It has units of ergs cm^{-3} Hz^{-1}. To get the *total radiation energy density*, we integrate over all frequencies to get

$$\mathcal{E}(\vec{x},t) = \frac{1}{c}\int_0^\infty \oint I(\vec{x},t,\vec{n},\nu)d\Omega d\nu \tag{2.6}$$

Total radiation energy density has units of ergs cm^{-3} [22].

2.3 Radiative Energy Flux

The first-order moment of the specific intensity—the specific intensity multiplied by propagation angle and then integrated over all solid angle—is referred to as the *monochromatic radiation flux*

$$\vec{F}(\vec{x},t,\nu) = \oint I(\vec{x},t,\vec{n},\nu)\vec{n}d\Omega \tag{2.7}$$

and has dimensions of ergs s^{-1} cm^{-2} Hz^{-1} [24].

To understand what the flux means physically, consider that the total number of photons crossing a differential surface element dS per frequency interval and unit time, and from all solid angles, is just the photon flux, $\mathcal{N}c\vec{n}$, integrated over all solid angles and dotted with the surface element

$$N = (\oint \mathcal{N}(\vec{x},t,\vec{n},\nu)c\vec{n}d\Omega)\cdot dS \tag{2.8}$$

This equation can be multiplied by the energy per photon, $h\nu$, to instead give a value for the total *energy* crossing dS.

$$Nh\nu = (\oint \mathcal{N}(\vec{x},t,\vec{n},\nu)ch\nu\vec{n}d\Omega)\cdot dS \tag{2.9}$$

Recalling the relationship between the specific intensity and the particle number density (equations 2.1 and 2.2), we can show that equation 2.9 is equivalent to the definition of monochromatic radiation flux given in equation 2.7 dotted into the differential surface dS. Therefore, we can say that the monochromatic radiation flux, $\vec{F}(\vec{x},t,\nu)$, is defined such that $\vec{F}\cdot dS$ gives the net rate of radiant energy flow through dS per unit time due to photons with frequency ν.

Another quantity of interest is the *integrated radiation flux* which is simply the monochromatic radiation flux integrated over all frequencies

$$\vec{F}(\vec{x},t) \equiv \int_0^\infty \vec{F}(\vec{x},t,\nu)d\nu \tag{2.10}$$

and has dimensions of ergs s^{-1} cm^{-2}. In English, the integrated radiation flux dotted into a differential surface dS is defined as the net rate of total radiant energy flow through dS.

2.4 Radiation Pressure Tensor

Taking the second-order moment of the specific intensity gives the *radiation pressure tensor*

$$\mathsf{P}(\vec{x},t,\nu) = \frac{1}{c}\oint I(\vec{x},t,\vec{n},\nu)\vec{n}\vec{n}d\Omega \tag{2.11}$$

the components of which have dimensions of dynes seconds cm^{-2} [22]. The physical meaning of this quantity is best revealed by first examining photon momentum. If a photon has energy $h\nu$, its momentum is $\frac{h\nu}{c}\vec{n}$. If we multiply the photon number density by the momentum per photon we get a *monochromatic radiation momentum density* vector which, via equations 2.1 and 2.2, is equal to

$$\vec{\mathcal{P}} = \frac{h\nu}{c}\vec{n}\mathcal{N}(\vec{x},t,\vec{n},\nu) \tag{2.12}$$

$$= \frac{1}{c^2}I(\vec{x},t,\vec{n},\nu)\vec{n} \tag{2.13}$$

Comparing equation 2.13 with equation 2.11 reveals an interesting similarity. Combining the two gives

$$\mathsf{P}(\vec{x}, t, \nu) = \oint \vec{\mathcal{P}} c\vec{n} d\Omega \tag{2.14}$$

The quantity $c\vec{n}$ is the photon velocity vector, so the physical meaning of the radiation pressure tensor is simply a means of describing the advection of momentum density. The specifics of what component of the momentum density is being advected in which direction is easily seen when we look at the radiation pressure tensor in component form, replacing the specific intensity with the photon number distribution via equations 2.1 and 2.2:

$$P_{ij}(\vec{x}, t, \nu) = \oint (\mathcal{N}(\vec{x}, t, \vec{n}, \nu) c n_i)(\frac{h\nu}{c} n_j) d\Omega \tag{2.15}$$

The above equation can be interpreted as the net flux of radiation momentum in the j direction per unit time due to photons at frequency ν through a unit area oriented perpendicular to the i direction. As it turns out, this is exactly the definition of *pressure* in a fluid which is why this quantity is referred to as the radiation pressure [22].

When the radiation field is isotropic, the radiation pressure tensor becomes diagonal and isotropic. In this case, it can be reduced to the average of the diagonal components of P, a scalar "hydrostatic" pressure which is referred to as the *mean radiation pressure*

$$\bar{P} \equiv \frac{1}{3}(P_{xx} + P_{yy} + P_{zz}) \tag{2.16}$$

Combining this with equation 2.11 gives the full form of the mean radiation pressure

$$\bar{P}(\vec{x}, t, \nu) = \frac{1}{3c} \oint I(\vec{x}, t, \vec{n}, \nu) d\omega \tag{2.17}$$

And this, in turn, can be combined with equation 2.2 to reveal

$$\bar{P}(\vec{x}, t, \nu) = \frac{1}{3} \oint \mathcal{E}(\vec{x}, t, \vec{n}, \nu) d\Omega \tag{2.18}$$

which is to say that the mean radiation pressure is one third of the monochromatic radiation energy density when the radiation field is isotropic [24].

Chapter 3

The Equation of Radiation Transfer

Now that we have developed a thorough description of the radiation field and some of the quantities derived from it, we can proceed to examining how the radiation field changes in time. The principal mechanisms for altering the radiation field are photon advection and interaction with the matter through which the photons are advecting. In the absence of matter interaction, photons move in straight lines at a constant speed of c. Their advection and its effects on the specific intensity can be described in purely mechanical terms and this will not be covered here. Rather, we will concentrate on the much more complicated photon-matter interactions.

3.1 Absorption, Scattering, and Total Opacities

As photons travel through matter, some number of them will be absorbed by the material and their energy will be lost from the radiation field. This loss of photon number density and energy density from the radiation field is described by a quantity called the *absorption opacity*, $\kappa_a(\vec{x}, t, \vec{n}, \nu)$. The absorption opacity is defined such that a beam of photons having specific intensity $I(\vec{x}, t, \vec{n}, \nu)$ moving through a piece of matter with differential cross section dS and thickness dl for a time interval dt will

lose an amount of energy given by

$$dE_a = \kappa_a(\vec{x}, t, \vec{n}, \nu) I(\vec{x}, t, \vec{n}, \nu) dl dS d\Omega d\nu dt \tag{3.1}$$

The absorption opacity is the product of an atomic absorption cross section (with units of cm^2) and the number density of absorbing atoms (with units of cm^{-3}) summed over all atomic states that can readily absorb photons of frequency ν. The units of κ_a are therefore cm^{-1} [24].

The other process that serves to remove energy from a beam of photons is scattering. Although scattering doesn't remove photons from the radiation field (the integral of $\mathcal{N}$ over all angles and frequencies remains the same), it does change the propagation angle and possibly the frequency of the scattered photon and therefore redistributes the photon elsewhere in phase space. As with absorption, scattering serves to remove energy from a beam of coherent photons. There is a corresponding *scattering opacity*, $\kappa_s(\vec{x}, t, \vec{n}, \nu)$, which is defined in the same way as κ_a except that the events leading to beam energy loss are scatters instead of absorption [24].

The absorption opacity and scattering opacity can be combined linearly to form what is called the *total opacity*, $\kappa_t(\vec{x}, t, \nu)$, because it is assumed that absorption and scattering events occur independently of each other.

$$\kappa_t(\vec{x}, t, \nu) \equiv \kappa_a(\vec{x}, t, \nu) + \kappa_s(\vec{x}, t, \nu) \tag{3.2}$$

The total opacity describes the tendency of a beam of photons to lose energy due to interactions with matter. The assumption is that scattering and absorption are the only two events that will cause a beam of photons to lose energy. The amount of energy lost can be written as

$$\begin{aligned} dE_t &= \kappa_t(\vec{x}, t, \vec{n}, \nu) I(\vec{x}, t, \vec{n}, \nu) dl dS d\Omega d\nu dt \\ &= [\kappa_a(\vec{x}, t, \vec{n}, \nu) + \kappa_s(\vec{x}, t, \vec{n}, \nu)] I(\vec{x}, t, \vec{n}, \nu) dl dS d\Omega d\nu dt \end{aligned} \tag{3.3}$$

The inverse of the total opacity is a description of the average distance over which a

photon can travel before undergoing either a scattering event or being absorbed into the material and is referred to as the *photon mean-free-path*, λ.

$$\lambda(\vec{x}, t, \vec{n}, \nu) = \frac{1}{\kappa_t(\vec{x}, t, \vec{n}, \nu)} \tag{3.4}$$

It is particularly important to this thesis to point out that the frequency variation of the above defined opacities can be exceedingly complicated. The structure of the opacity vs. frequency spectrum is a result of contributions from virtually countless atomic transitions (bound-bound, bound-free, and free-free) and the changes in atomic interaction cross section that occur over each transition.

3.2 Emission

The remaining major interaction event between photons and matter is the emission of photons by atoms. As a beam of photons travels through a medium, some atomic emissions could have an appropriate propagation angle, frequency, position, and time such that they effectively add to the specific intensity of the beam. This phenomenon is described by a quantity called the *emissivity*, $\eta(\vec{x}, t, \vec{n}, \nu)$.

The emissivity is defined such that the amount of radiant energy released by a material element of cross section dS and length dl into solid angle $d\Omega$ around direction $\vec{n}$ with frequencies between ν and $\nu + d\nu$ in a time interval dt is

$$dE_\eta = \eta(\vec{x}, t, \vec{n}, \nu) dl dS d\Omega d\nu dt \tag{3.5}$$

Emissivity has units of ergs cm^{-3} sr^{-1}. It is calculated by summing the products of atomic state populations and transition probabilities over all processes that can result in the emission of a photon at frequency ν [22].

It is important to understand that photons entering a particular point in phase space $(\vec{x}, t, \vec{n}, \nu)$ via thermal emission from atoms are indistinguishable from photons entering the same point in phase space as a result of a scattering event. Therefore, we break the total emissivity (η_t from here on) up into two components—the thermal

emissivity, η_T, and the scattering emissivity, η_s. The thermal and scattering components of total emissivity are independent of each other and can be added linearly.

3.3 The Transfer Equation

At this point we understand the radiation field and the various material—photon interactions that can alter it. The next step is to derive an equation which describes the time dynamics of the radiation field and incorporates all of the processes described above. This equation is called the *radiation transfer equation.*

Consider an element of material with differential cross section dS and thickness dl. In a time interval dt, we are interested in the change in the specific intensity of a beam of photons with frequencies between ν and $\nu + d\nu$, traveling into solid angle $d\Omega$ along a direction $\vec{n}$ normal to dS. The difference in the specific intensity between $(\vec{x}, t)$ and $(\vec{x}+\Delta\vec{x}, t+\Delta t)$ must be a result of energy lost to scattering and absorption and energy gained through emission. This can be written as

$$\begin{aligned} &[I(\vec{x}+\Delta\vec{x}, t+\Delta t, \vec{n}, \nu) - I(\vec{x}, t, \vec{n}, \nu)]dSd\Omega d\nu dt \\ &= [\eta_t(\vec{x}, t, \vec{n}, \nu) - \kappa_t(\vec{x}, t, \vec{n}, \nu)I(\vec{x}, t, \vec{n}, \nu)]dldSd\Omega d\nu dt \end{aligned} \tag{3.6}$$

and simplified to the standard form of the radiation transfer equation

$$[\frac{1}{c}\frac{\partial}{\partial t} + (\vec{n}\cdot\vec{\nabla})]I(\vec{x}, t, \vec{n}, \nu) = \eta_t(\vec{x}, t, \vec{n}, \nu) - \kappa_t(\vec{x}, t, \vec{n}, \nu)I(\vec{x}, t, \vec{n}, \nu)\,. \tag{3.7}$$

The left-hand side of this equation is what is sometimes referred to as the "convective" derivative of specific intensity. A convective derivative is a derivative taken with respect to a moving coordinate system, also called the Lagrangian derivative. It takes into account all changes in specific intensity as a result of local sources and sinks of photons, as well as contributions from photons moving in and out of the local area. Separating the left-hand side of this equation allows us to write the transfer

equation as a conservation equation:

$$\begin{aligned}\frac{\partial}{\partial t}I(\vec{x},t,\vec{n},\nu) &= c\eta_t(\vec{x},t,\vec{n},\nu) - c\kappa_t(\vec{x},t,\vec{n},\nu)I(\vec{x},t,\vec{n},\nu) \\ &\quad -c(\vec{n}\cdot\vec{\nabla})I(\vec{x},t,\vec{n},\nu) \end{aligned} \tag{3.8}$$

In this form, the physical meaning of the transfer equation is obvious. The change in time of specific intensity is equal to the photon sources minus the photons sinks. In this case, the sources of photons are given by the total emissivity plus the photons that enter position $\vec{x}$ due to advection, while the photon sinks are comprised of the total opacity times the specific intensity and the photons that leave position $\vec{x}$ due to advection.

3.4 Moments of the Transfer Equation

Just as the moments of the specific intensity revealed useful physical quantities, so will moments of the transfer equation. To obtain the zeroth order moment equation, we integrate equation 3.7 over all angles

$$\begin{aligned}\oint \frac{1}{c}\frac{\partial}{\partial t}I(\vec{x},t,\vec{n},\nu)d\Omega + \oint(\vec{n}\cdot\vec{\nabla})I(\vec{x},t,\vec{n},\nu)d\Omega = \\ \oint[\eta_t(\vec{x},t,\vec{n},\nu) - \kappa_t(\vec{x},t,\vec{n},\nu)I(\vec{x},t,\vec{n},\nu)]d\Omega \end{aligned} \tag{3.9}$$

The first term can be written as a function of the radiation energy density by combining it with equation 2.5. Similarly, the second term can be combined with equation 2.7 to be written in terms of the monochromatic radiation flux. These two manipulations produce the following form of the zeroth order moment equation:

$$\frac{\partial}{\partial t}\mathcal{E}(\vec{x},t,\nu) + \vec{\nabla}\cdot\vec{F}(\vec{x},t,\nu) = \oint[\eta_t(\vec{x},t,\vec{n},\nu) - \kappa_t(\vec{x},t,\vec{n},\nu)I(\vec{x},t,\vec{n},\nu)]d\Omega \tag{3.10}$$

In this form, the zeroth order moment equation is a conservation equation for radiation energy density. The terms are easily described physically. The time rate of change in monochromatic radiation energy density at frequency ν is equal to the sum of all

energy emitted at frequency ν by the material, minus the sum of all energy in photons of frequency ν absorbed or scattered by the material, minus the net flow of energy in photons of frequency ν through the surface containing the volume element (the $\vec{\nabla} \cdot \vec{F}$ term).

The first-order moment equation for the radiation field is obtained by multiplying the transfer equation (3.7) by $\vec{n}$ and integrating over all angles. By incorporating equations 2.7 and 2.11, this treatment produces the following form of the first-order moment equation

$$\frac{1}{c^2}\frac{\partial}{\partial t}\vec{F}(\vec{x},t,\nu)+\vec{\nabla}\cdot\mathsf{P}(\vec{x},t,\nu) = \frac{1}{c}\oint[\eta_t(\vec{x},t,\vec{n},\nu)-\kappa_t(\vec{x},t,\vec{n},\nu)I(\vec{x},t,\vec{n},\nu)]\vec{n}d\Omega \quad (3.11)$$

Substituting equations 2.7 and 2.13 into the first term of equation 3.11 reveals that this is just the time derivative of the radiation momentum density integrated over all propagation angles. Recall also from equation 2.15 that P is the radiation momentum flux density tensor. Clearly, then, the first moment of the radiation transfer equation is effectively a conservation equation for the radiation momentum density and is analogous to the hydrodynamical equations of motion.

The physical interpretation of equation 3.11 is that the time rate of change of monochromatic radiation momentum density for photons at frequency ν is equal to the negative of the volumetric force exerted by radiation stresses caused by photons at frequency ν plus the net momentum gained by the photons of frequency ν in the radiation field from interactions with the material.

3.5 Local Thermal Equilibrium

The radiation transfer equation and its moments are reasonably complicated and are difficult to solve in any sort of general sense. There exist, however, certain cases for which the transfer equation is greatly simplified. Local thermal equilibrium (LTE) is one such case.

Defining LTE requires that we first define thermodynamic equilibrium (TE). An

adiabatic system is said to be in strict TE if it is in steady-state equilibrium and contains a homogeneous material. In this case, the material must have the same temperature everywhere—otherwise temperature gradients would exist, inducing heat flow and the loss of steady-state. For the same reason, it is safe to assume that the radiation field is isotropic and homogeneous. Furthermore, because the material is exhibiting blackbody radiation (and thereby losing energy), the rate of photon absorption by the material must be such that the energy gained by the material by said absorption exactly cancels the blackbody radiation. The mathematical version of this statement is

$$\eta_T(\nu) = \kappa_a(\nu) I(\vec{n}, \nu) \tag{3.12}$$

and is known as *Kirchoff's law.*

Material in strict TE at temperature T has specific intensity described by the *Planck function*

$$B(\nu, T) = \frac{2h\nu^3}{c^2}(e^{\frac{h\nu}{kT}} - 1)^{-1} \tag{3.13}$$

where k is the *Boltzmann constant*, $1.3806503 \times 10^{-23}$ m^2 kg s^{-2} K^{-1}. Equation 3.13 combined with equation 3.12 leads to the *Kirchhoff-Planck relation* [22]

$$\eta_T(\nu) = \kappa_a(\nu) B(\nu, T) \tag{3.14}$$

The Kirchhoff-Planck relation only holds true for strict TE. However, if the thermal gradients in the material are so small that the change in temperature over the mean-free-path of a photon were negligible, then we can assume that equation 3.14 will offer a good approximation at the local values for the thermodynamic variables. The reasoning is that a photon will be thermalized by the material long before it sees a significant difference in temperature, so in the region through which the photon streams, the thermodynamic variables are essentially constant and a "local steady-state" can be assumed. This is the fundamental hypothesis of LTE.

In the case of LTE, we can rewrite the Kirchhoff-Planck relation to include the

local values of position and time.

$$\eta_T(\vec{x},t,\nu) = \kappa_a(\vec{x},t,\nu)B[\nu,T(\vec{x},t)] \tag{3.15}$$

Furthermore, if we assume that the scattering that takes place in the system is coherent and isotropic, then we can write the scattering emissivity as

$$\eta_s = \frac{\kappa_s}{4\pi}\oint I d\Omega \tag{3.16}$$

and therefore the entire emissivity can be written as

$$\eta_t = \kappa_a(\vec{x},t,\nu)B[\nu,T(\vec{x},t)] + \frac{\kappa_s(\vec{x},t,\nu)}{4\pi}\oint I(\vec{x},\vec{n},t,\nu)d\Omega \tag{3.17}$$

3.6 The Diffusion Limit

In regions of high optical depth and low photon mean-free-path, photons interact frequently and have a tendency to undergo many interactions before traveling a distance comparable to the scale lengths of interest. Photons "diffuse" through matter in a random walk of scattering events until they are absorbed. High electron density and high temperature makes for frequent electron scattering events which rapidly drive the electron energy spectrum to equilibrium. Any effect on the material energy by the coupling to the radiation energy field is equilibrated effectively instantaneously. This environment supports a condition of LTE [23].

The principal assumption of the diffusion approximation is that the photon mean free path is much shorter than the scale lengths of the problem, i.e. $\lambda \ll l$. In the limit that this is true, a perturbation analysis of the radiation transfer equation provides a rigorous and elegant derivation of the diffusion limit of transport. To begin, we start with the transfer equation (3.7) and rewrite it in the following form

$$\frac{1}{c}\frac{\partial I}{\partial t} + (\vec{n}\cdot\vec{\nabla})I = \kappa_a(B-I) - S(I) \tag{3.18}$$

where $S(I)$ is the scattering operator allowing for photon scattering in angle and frequency. Here we have made the assumption of LTE, which dictates that the thermal emission spectrum is always Planckian and the thermal emissivity can therefore be written as $\kappa_a B(T_m)$ where T_m is the material temperature. The LTE assumption also simplifies the scattering operator as will be seen shortly. Unlike strict TE, however, LTE allows for the material and photon temperatures to differ. In the limit that λ/l is small, the material and radiation temperatures will be similar. We will assume for the purposes of this derivation that the matter temperature, T_m, and the radiation temperature, T_r, are similar such that the energy exchange between the radiation field and matter is slow enough to compete with transport, i.e. $(T_m - T_r)/T_r \ll 1$.

In the diffusion limit of small λ/l, we can say that $T_m - T_r$ is also small. Writing equation 3.18 as

$$\frac{1}{c}\frac{\partial I}{\partial t} + (\vec{n}\cdot\vec{\nabla})I = \kappa_a(B(T_r) - I + B(T_m) - B(T_r)) - S(I)\,. \tag{3.19}$$

we can treat $B(T_m) - B(T_r)$ as small. This is the matter-radiation energy exchange term.

The Scattering Operator

We now turn to consider the properties of the scattering operator, S, on the specific intensity, I. We start from the form of the operator (for Compton scattering of photons off free electrons) used by Sampson[39]. We express the operator in terms of the photon distribution function to help elucidate the physics and to simplify proof of the operator's conservation law properties.

Sampson's formulation for $S(I)$ is

$$S(I) = \oint\Big\{\int_0^\infty N(p_e,T_m)dp_e\sigma(\nu\to\nu',\Omega\to\Omega')I(\nu,\Omega)[1+\frac{c^2}{2h\nu'^3}I(\nu',\Omega')] - \int_0^\infty N(p'_e,T_m)dp'_e\sigma(\nu'\to\nu,\Omega'\to\Omega)\frac{\nu}{\nu'}\frac{d\nu'}{d\nu}I(\nu',\Omega')[1+\frac{c^2}{2h\nu^3}I(\nu,\Omega)]\Big\}d\Omega' \tag{3.20}$$

where $\sigma(\nu\to\nu',\Omega\to\Omega')$ is the differential scattering cross section for photons

scattering from initial frequency ν to final frequency ν' and from initial propagation angle Ω to final angle Ω'. $N(p_e, T_m)$ is the number density of electrons with momentum p_e.

We will rewrite this using the following definition

$$I = h\nu\rho(\nu)n_\nu \tag{3.21}$$

where n_ν is the distribution function for photons at frequency ν and $\rho(\nu)$ is the density of states for photons with frequency ν

$$\rho(\nu) \equiv \frac{2\nu^2}{c^2}\,. \tag{3.22}$$

Rewriting the scattering operator as a function of n_ν leads to

$$S(n_\nu) = \oint\Big\{\int_0^\infty N(p_e, T_m)dp_e\sigma(\nu \to \nu', \Omega \to \Omega')\frac{2h\nu^3}{c^2}n_\nu(1+n_{\nu'}) - \int_0^\infty N(p_e', T_m)dp_e'\sigma(\nu' \to \nu, \Omega' \to \Omega)\frac{\nu}{\nu'}\frac{d\nu'}{d\nu}\frac{2h\nu'^3}{c^2}n_{\nu'}(1+n_\nu)\Big\}d\Omega'\,. \tag{3.23}$$

This can be written in a more symmetric form by reintroducing the frequency integral and associated energy conservation delta function

$$1 \to \int_0^\infty d\nu\delta\Big(\nu - \nu' - \frac{E_e - E_e'}{h}\Big) \tag{3.24}$$

along with the principle of detailed balance, which tells us that the probability of transition from (ν, Ω) to (ν', Ω') is equal to the probability of the inverse process. Sampson writes the principle of detailed balance like this

$$\sigma(\nu \to \nu', \Omega \to \Omega')\nu^2 d\nu dp_e = \sigma(\nu' \to \nu, \Omega' \to \Omega)\nu'^2 d\nu' dp_e'\,. \tag{3.25}$$

If we define the following shorthand for the transition rates

$$\mathcal{T}_{\nu\to\nu'} \;=\; \sigma(\nu \to \nu', \Omega \to \Omega')dp_e c\rho(\nu) \tag{3.26}$$

$$\mathcal{T}_{\nu'\to\nu} = \sigma(\nu'\to\nu,\Omega'\to\Omega)dp'_e c\rho(\nu') \tag{3.27}$$

$$\tag{3.28}$$

we can express the scattering operator as

$$\begin{aligned} S(n_\nu) = & \frac{2}{c^2}\oint d\Omega' \int_0^\infty d\nu' \int_0^\infty dp_e \\ & \times h\nu \mathcal{T}_{\nu\to\nu'} N(p_e,T_m) n_\nu (1+n_{\nu'}) \delta(\nu-\nu'-\frac{1}{h}(E_e-E'_e)) \\ & -\frac{2}{c^2}\oint d\Omega' \int_0^\infty d\nu' \int_0^\infty dp'_e \\ & \times h\nu \mathcal{T}_{\nu'\to\nu} N(p'_e,T_m) n_{\nu'} (1+n_\nu) \delta(\nu-\nu'-\frac{1}{h}(E_e-E'_e)) \,. \end{aligned} \tag{3.29}$$

These two terms represent the amount of energy lost and gained by the intensity, $I = h\nu\rho n_\nu$, as the result of scattering.

We can split up the energy conservation delta function into a portion which assumes fully elastic Thompson scattering and a term which accounts for inelastic Compton energy exchange by incorporating this equality:

$$\delta(\nu-\nu'-\frac{1}{h}(E_e-E'_e)) = \delta(\nu-\nu') + \left[\delta(\nu-\nu'-\frac{1}{h}(E_e-E'_e)) - \delta(\nu-\nu')\right] \tag{3.30}$$

where we assume that $\delta(\nu-\nu')$ is leading order while $[\delta(\nu-\nu'-\frac{1}{h}(E_e-E'_e))-\delta(\nu-\nu')]$ is a second order contribution. Substituting this equality into equation 3.29 allows us to write the following

$$S(n_\nu) = S^{elastic}(n_\nu) + S^{Compton}(n_\nu) \tag{3.31}$$

where

$$\begin{aligned} S^{elastic}(n_\nu) = & \frac{2}{c^2}\oint d\Omega' \int_0^\infty d\nu' \int_0^\infty dp_e \times \\ & h\nu \mathcal{T}_{\nu\to\nu'} N(p_e,T_m) n_\nu (1+n_{\nu'}) \delta(\nu-\nu') - \\ & \frac{2}{c^2}\oint d\Omega' \int_0^\infty d\nu' \int_0^\infty dp'_e \times \\ & h\nu \mathcal{T}_{\nu'\to\nu} N(p'_e,T_m) n_{\nu'} (1+n_\nu) \delta(\nu-\nu') \,. \end{aligned} \tag{3.32}$$

and

$$
\begin{aligned}
S^{Compton}(n_\nu) &= \frac{2}{c^2}\oint d\Omega' \int_0^\infty d\nu' \int_0^\infty dp_e h\nu \mathcal{T}_{\nu\to\nu'} N(p_e, T_m) n_\nu (1+n_{\nu'}) \times \\
&\quad \delta(\nu - \nu' - \frac{1}{h}(E_e - E_e')) - \\
&\quad \frac{2}{c^2}\oint d\Omega' \int_0^\infty d\nu' \int_0^\infty dp_e' h\nu \mathcal{T}_{\nu'\to\nu} N(p_e', T_m) n_{\nu'} (1+n_\nu) \times \\
&\quad \delta(\nu - \nu' - \frac{1}{h}(E_e - E_e')) - S^{elastic}(n_\nu)\,. \qquad (3.33)
\end{aligned}
$$

The energy conservation properties of $S^{elastic}(n_\nu)$ can be shown by integrating over all angles and photon frequencies.

$$
\begin{aligned}
\oint d\Omega \int_0^\infty d\nu S^{elastic}(n_\nu) &= \frac{2}{c^2}\oint d\Omega \int_0^\infty d\nu \oint d\Omega' \int_0^\infty d\nu' \int_0^\infty dp_e \times \\
&\quad h\nu \mathcal{T}_{\nu\to\nu'} N(p_e, T_m) n_\nu (1+n_{\nu'}) \delta(\nu - \nu') - \\
&\quad \frac{2}{c^2}\oint d\Omega \int_0^\infty d\nu \oint d\Omega' \int_0^\infty d\nu' \int_0^\infty dp_e' \times \\
&\quad h\nu \mathcal{T}_{\nu'\to\nu} N(p_e', T_m) n_\nu' (1+n_\nu) \delta(\nu - \nu')\,. \qquad (3.34)
\end{aligned}
$$

By swapping which frequency we denote with the prime in the second term, we can show that the two terms are equal and cancel each other. The result is, as expected, that the total change in energy due to Thomson scattering is zero. In equation 3.30, we assumed that the elastic scattering component was leading order while the Compton contribution was a second-order correction. This indicates that the complete scattering operator conserves energy to first order. We can write this in terms of specific intensity

$$
\oint d\Omega \int_0^\infty d\nu S^{elastic}(I) = 0\,. \qquad (3.35)
$$

The fact that scattering conserves energy to first order for any specific intensity is significant and will play an important role in the perturbation analysis, below.

The scattering operator drives the radiation intensity to the blackbody distribution at the matter temperature. In equilibrium, with no transport, the absorption of photons out of the radiation field is exactly matched by the emissivity of the

material. In order to maintain a steady state, the in-scatter and out-scatter from any point in specific intensity phase space must also exactly cancel. This mandates a fully-balanced scattering operator and we can say that $S(I) = 0$. The balanced nature of the scattering operator in equilibrium allows us to reduce equation 3.29 to

$$N(\vec{p_e}, T_m)n_\nu(1 + n_{\nu'}) = N(\vec{p_e'}, T_m)n_{\nu'}(1 + n_\nu)\,. \tag{3.36}$$

If we now define a substitution variable

$$\hat{n_\nu} = \frac{n_\nu}{1 + n_\nu} \tag{3.37}$$

we can rewrite equation 3.36 as

$$\frac{\hat{n_\nu}}{\hat{n_{\nu'}}} = \frac{N(\vec{p_e}, T_m)}{N(\vec{p_e'}, T_m)}\,. \tag{3.38}$$

The factor on the right hand side is described by the Saha-Boltzmann equation, which allows us to write

$$\frac{\hat{n_\nu}}{\hat{n_{\nu'}}} = \frac{\exp(-\frac{h\nu}{kT_m})}{\exp(-\frac{h\nu'}{kT_m})}\,. \tag{3.39}$$

Separating the variables here shows that

$$\hat{n_\nu} = \exp(-\frac{h\nu}{kT_m}) \tag{3.40}$$

and therefore, via equation 3.37,

$$n_\nu = \frac{1}{\exp(-\frac{h\nu}{kT_m}) - 1}\,. \tag{3.41}$$

Substituting this and equation 3.22 into equation 3.21 shows that

$$I = \frac{2h\nu^3}{c^2}\frac{1}{\exp(\frac{h\nu}{kT_m}) - 1}\,. \tag{3.42}$$

Comparing this equation to 3.13 shows that this is the definition of the Planck dis-

tribution. Starting with the statement that we are in equilibrium, we are able to conclude that $I = B$ and furthermore that $S(I) = 0$. Therefore, we can say that

$$S(B) = 0\,. \tag{3.43}$$

Perfect balance of the scattering operator is only true when in equilibrium, so we can also say that if $S(I) = 0$ then I *must* be B.

In equation 3.19 we split the temperatures such that we could account for the possible difference between T_m and T_r. The result is a difference term, which is assumed to be small, and a dominant T_r term. In order to write the complete transfer equation with the same dominate temperature, we must also split the electron number density term in the scattering operator in the same way

$$N(\vec{p}_e, T_m) = N(\vec{p}_e, T_r) + N(\vec{p}_e, T_m) - N(\vec{p}_e, T_r)\,. \tag{3.44}$$

where we can treat $N(\vec{p}_e, T_m) - N(\vec{p}_e, T_R)$ as small.

Perturbation Analysis

Now we take the rewritten transfer equation and expand the specific intensity in powers of $\epsilon = \lambda/l$:

$$I = I_0 + I_1 + I_2 + ... \tag{3.45}$$

In this way, we expand about thermal equilibrium and ensure that all of the frequency- and angle-integrated specific intensity is contained in I_0. To construct an ordering for the operators in the radiation transfer equation, we start with the collision term, which goes as $1/\lambda$, and assign it an order of 1. In order to make it dimensionless, we must multiply by a factor of λ. The advection term goes as $1/l$, so when multiplied by λ it goes as $\lambda/l = \epsilon$. We assign the $(B(T_m) - B(T_r))$ term in the energy exchange operator to be order ϵ^2 on account of the assumption that $(T_m - T_r)/T_r \ll 1$. The time derivative term is assigned an order of ϵ^2 because this is the only ordering which allows the specific intensity to go to a Planckian to leading order. We can now write

out the order of each operator in the transfer equation

$$\begin{aligned} \mathcal{C} &= \mathcal{O}(1) && (3.46)\\ (\vec{n}\cdot\vec{\nabla}) &= \mathcal{O}(\epsilon) && (3.47)\\ \frac{1}{c}\frac{\partial}{\partial t} &= \mathcal{O}(\epsilon^2) && (3.48)\\ \mathcal{X} &= \mathcal{O}(\epsilon^2) && (3.49) \end{aligned}$$

where we have used the $\mathcal{C}$ notation to denote the non-linear collision operator which includes both absorption and scattering terms

$$\mathcal{C}(I) = \kappa_a(B(T_r) - I) - S(I) \tag{3.50}$$

and the $\mathcal{X}$ notation to denote the exchange term, which handles the effects on absorption due to a difference between material and photon temperatures

$$\mathcal{X} = \kappa_a(B(T_m) - B(T_r))\,. \tag{3.51}$$

All of these operators are linear in specific intensity except for the collision operator. As a result, simply applying the collision operator on I_2 does not produce the full second-order collision terms. There are nonlinear terms in I_1 as well as I_0 terms combined with a second-order representation of electron number density or the second-order Compton energy delta function. In order to perform a proper perturbation analysis, we must first expand $\mathcal{C}(I)$ and collect terms by order.

To begin, we substitute equation 3.45 into equation 3.50 to obtain

$$\mathcal{C}(I_0 + I_1 + I_2) = \kappa_a(B(T_r) - I_0 - I_1 - I_2) - S(I_0 + I_1 + I_2)\,. \tag{3.52}$$

The non-scattering terms can be ordered immediately:

$$\begin{aligned} \kappa_a B(T_r) &= \mathcal{O}(1) && (3.53)\\ \kappa_a I_0 &= \mathcal{O}(1) && (3.54) \end{aligned}$$

$$\kappa_a I_1 = \mathcal{O}(\epsilon) \tag{3.55}$$

$$\kappa_a I_2 = \mathcal{O}(\epsilon^2) \tag{3.56}$$

Next, we expand the scattering term $S(I_0 + I_1 + I_2)$ and order the results. To do so, we need to take into account the temperature splitting in the electron number density, as described by equation 3.44, and the splitting of the elastic and inelastic scattering contributions described by equation 3.30. We assert that the quantity $(T_m - T_R)$ is $\mathcal{O}(\epsilon^2)$. Therefore, we can treat the term $N(p_e, T_r)$ as $\mathcal{O}(1)$ and the quantity $(N(p_e, T_m) - N(p_e, T_r))$ as $\mathcal{O}(\epsilon^2)$. The elastic delta function $\delta(\nu - \nu')$ is treated as $\mathcal{O}(1)$ while the Compton delta function, $\delta(\nu - \nu' - (E - E')/h) - \delta(\nu - \nu')$ is treated as $\mathcal{O}(\epsilon^2)$. These orderings will allow all of the energy exchange terms to compete at second order.

We are only keeping terms up to second order, so the scattering terms that include the second-order electron number density only need to have the zeroth-order specific intensity included. As this expansion is relatively simple, we'll handle it first. Separating the scattering operator on the two halves of the second-order electron number density gives us

$$\begin{aligned}
S_2^{enex}(I_0) = \oint \Big\{ & \int_0^\infty d\nu' \int_0^\infty N(p_e, T_m) dp_e \sigma(\nu \to \nu', \Omega \to \Omega') \times \\
& I_0[1 + \frac{c^2}{2h\nu'^3} I_0'] \delta(\nu - \nu') - \\
& \int_0^\infty d\nu' \int_0^\infty N(p_e', T_m) dp_e' \sigma(\nu' \to \nu, \Omega' \to \Omega) \frac{\nu}{\nu'} \frac{d\nu'}{d\nu} \times \\
& I_0'[1 + \frac{c^2}{2h\nu^3} I_0] \delta(\nu - \nu') - \\
& \int_0^\infty d\nu' \int_0^\infty N(p_e, T_r) dp_e \sigma(\nu \to \nu', \Omega \to \Omega') \times \\
& I_0[1 + \frac{c^2}{2h\nu'^3} I_0'] \delta(\nu - \nu') + \\
& \int_0^\infty d\nu' \int_0^\infty N(p_e', T_r) dp_e' \sigma(\nu' \to \nu, \Omega' \to \Omega) \frac{\nu}{\nu'} \frac{d\nu'}{d\nu} \times \\
& I_0'[1 + \frac{c^2}{2h\nu^3} I_0] \delta(\nu - \nu') \Big\} d\Omega'
\end{aligned} \tag{3.57}$$

where we have introduced the shorthand I_0 to indicate $I_0(\nu, \Omega)$ and I_0' to indicated

$I_0(\nu', \Omega')$. The notation $S_2^{enex}(I_0)$ indicates that this is a second-order term which is a non-linear function of I_0 and arrises as a result of the energy exchange temperature splitting in the electron number density. The terms in $S_2^{enex}(I_0)$ all contain the elastic scattering delta function, and are thus part of the Thomson scattering contribution to total scattering at second order. Physically, $S_2^{enex}(I_0)$ is just a second-order correction term which arises as a result of our choice to use $N(p_e, T_r)$ to leading order rather than $N(p_e, T_m)$.

In a similar fashion, the scattering terms that include the second-order Compton energy exchange delta function only need the zeroth-order specific intensity included.

$$\begin{aligned} S_2^{Compton}(I_0) \;=\; & \oint \Big\{ \int_0^\infty d\nu \int_0^\infty N(p_e, T_r) dp_e \sigma(\nu \to \nu', \Omega \to \Omega') \times \\ & I_0[1 + \frac{c^2}{2h\nu'^3} I_0'] \delta(\nu - \nu' - (E_e - E_e')/h) - \\ & \int_0^\infty d\nu \int_0^\infty N(p_e', T_r) dp_e' \sigma(\nu' \to \nu, \Omega' \to \Omega) \frac{\nu}{\nu'} \frac{d\nu'}{d\nu} \times \\ & I_0'[1 + \frac{c^2}{2h\nu^3} I_0] \delta(\nu - \nu' - (E_e - E_e')/h) - \\ & \int_0^\infty d\nu \int_0^\infty N(p_e, T_r) dp_e \sigma(\nu \to \nu', \Omega \to \Omega') \times \\ & I_0[1 + \frac{c^2}{2h\nu'^3} I_0'] \delta(\nu - \nu') - \\ & \int_0^\infty d\nu \int_0^\infty N(p_e', T_r) dp_e' \sigma(\nu' \to \nu, \Omega' \to \Omega) \frac{\nu}{\nu'} \frac{d\nu'}{d\nu} \times \\ & I_0'[1 + \frac{c^2}{2h\nu^3} I_0] \delta(\nu - \nu') \Big\} d\Omega' \end{aligned} \tag{3.58}$$

We can now write the total second-order expansion of the scattering operator as the linear combination of terms:

$$S(I_0 + I_1 + I_2) = S_0(I_0) + S_1(I_1) + S_2^{lin}(I_2) + S_2^{nl}(I_1) + S_2^{Compton}(I_0) + S_2^{enex}(I_0)\,. \tag{3.59}$$

where the subscripts on each term indicate its order, $S_2^{lin}(I_2)$ indicates a second order term that is linear in I_2, and $S_2^{nl}(I_1)$ indicates a second order term that is non-linear in I_1. The terms which have not yet been explicitly defined can be by doing the full expansion and separating the terms:

$$
\begin{aligned}
S_0(I_0) &= \oint\Big\{\int_0^\infty d\nu \int_0^\infty N(p_e,T_r)dp_e\sigma(\nu\to\nu',\Omega\to\Omega')\times \\
&\quad I_0[1+\frac{c^2}{2h\nu'^3}I_0']\delta(\nu-\nu') - \\
&\quad \int_0^\infty d\nu \int_0^\infty N(p_e',T_r)dp_e'\sigma(\nu'\to\nu,\Omega'\to\Omega)\frac{\nu}{\nu'}\frac{d\nu'}{d\nu}\times \\
&\quad I_0'[1+\frac{c^2}{2h\nu^3}I_0]\delta(\nu-\nu')\Big\}d\Omega' \qquad (3.60)
\end{aligned}
$$

$$
\begin{aligned}
S_1(I_1) &= \oint\Big\{\int_0^\infty d\nu \int_0^\infty \Big[N(p_e,T_r)dp_e\sigma(\nu\to\nu',\Omega\to\Omega')I_1 + \\
&\quad \frac{c^2}{2h\nu'^3}[I_1I_0' + I_1'I_0]\delta(\nu-\nu')\Big] - \\
&\quad \int_0^\infty d\nu \int_0^\infty \Big[N(p_e',T_r)dp_e'\sigma(\nu'\to\nu,\Omega'\to\Omega)\frac{\nu}{\nu'}\frac{d\nu'}{d\nu}I_1' + \\
&\quad \frac{c^2}{2h\nu^3}[I_1'I_0 + I_1I_0']\delta(\nu-\nu')\Big]\Big\}d\Omega' \qquad (3.61)
\end{aligned}
$$

$$
\begin{aligned}
S_2^{lin}(I_2) &= \oint\Big\{\int_0^\infty d\nu \int_0^\infty \Big[N(p_e,T_r)dp_e\sigma(\nu\to\nu',\Omega\to\Omega')I_2 + \\
&\quad \frac{c^2}{2h\nu'^3}[I_2I_0' + I_2'I_0]\delta(\nu-\nu')\Big] - \\
&\quad \int_0^\infty d\nu \int_0^\infty \Big[N(p_e',T_r)dp_e'\sigma(\nu'\to\nu,\Omega'\to\Omega)\frac{\nu}{\nu'}\frac{d\nu'}{d\nu}I_2' + \\
&\quad \frac{c^2}{2h\nu^3}[I_2'I_0 + I_2I_0']\delta(\nu-\nu')\Big]\Big\}d\Omega' \qquad (3.62)
\end{aligned}
$$

$$
\begin{aligned}
S_2^{nl}(I_1) &= \oint\Big\{\int_0^\infty d\nu \int_0^\infty N(p_e,T_r)dp_e\sigma(\nu\to\nu',\Omega\to\Omega')\times \\
&\quad \frac{c^2}{2h\nu'^3}I_1I_1'\delta(\nu-\nu') - \\
&\quad \int_0^\infty d\nu \int_0^\infty N(p_e',T_r)dp_e'\sigma(\nu'\to\nu,\Omega'\to\Omega)\times
\end{aligned}
$$

$$\frac{\nu}{\nu'}\frac{d\nu'}{d\nu}\frac{c^2}{2h\nu^3}I_1' I_1 \delta(\nu-\nu')\Big\} d\Omega' \tag{3.63}$$

Notice that $S_0(I_0)$ is identical to the second half of $S_2^{enex}(I_0)$ so we can simplify equation 3.57 to

$$\begin{aligned} S_2^{enex}(I_0) &= \oint\Big\{\int_0^\infty d\nu \int_0^\infty N(p_e,T_m)dp_e\sigma(\nu\to\nu',\Omega\to\Omega')\times \\ &\quad I_0[1+\frac{c^2}{2h\nu'^3}I_0'] - \\ &\quad \int_0^\infty d\nu\int_0^\infty N(p_e',T_m)dp_e'\sigma(\nu'\to\nu,\Omega'\to\Omega)\times \\ &\quad \frac{\nu}{\nu'}\frac{d\nu'}{d\nu}I_0'[1+\frac{c^2}{2h\nu^3}I_0] - S_0(I_0)\Big\}d\Omega'\,. \end{aligned} \tag{3.64}$$

$S_0(I_0)$ also appears as the second half of $S_2^{Compton}(I_0)$, so we can simplify equation 3.58 to

$$\begin{aligned} S_2^{Compton}(I_0) &= \oint\Big\{\int_0^\infty d\nu \int_0^\infty N(p_e,T_r)dp_e\sigma(\nu\to\nu',\Omega\to\Omega')\times \\ &\quad I_0[1+\frac{c^2}{2h\nu'^3}I_0']\delta(\nu-\nu'-(E_e-E_e')/h) - \\ &\quad \int_0^\infty d\nu\int_0^\infty N(p_e',T_r)dp_e'\sigma(\nu'\to\nu,\Omega'\to\Omega)\frac{\nu}{\nu'}\frac{d\nu'}{d\nu}\times \\ &\quad I_0'[1+\frac{c^2}{2h\nu^3}I_0]\delta(\nu-\nu'-(E_e-E_e')/h) - S_0(I_0)\Big\}d\Omega' \end{aligned} \tag{3.65}$$

Now that we have expanded the scattering operator and ordered the terms, we can assemble the full transfer equation by order and examine the results.

Zeroth Order

If we take only the zeroth-order terms in the transfer equation we are left with

$$0 = \kappa_a(B(T_r)-I_0) - S_0(I_0)\,. \tag{3.66}$$

As we are expanding about thermal equilibrium, we expect the zeroth-order behavior of the radiation field to be similar to that of strict thermal equilibrium. In strict TE, scattering is fully balanced and thus we expect $S_0(I_0)$ to be zero. If this

is the case, then the above equation reduces to $B(T_r) = I_0$. Substituting this result back into equation 3.66 leaves us with $S_0(B(T_r)) = 0$. This expression was already proven to be true in equation 3.43, leaving us with a fully self-consistent result. It is also consistent with the assertion that the zeroth-order behavior of the transfer equation in the diffusion limit is equivalent to strict TE, where the specific intensity is always represented by a Planckian. The zeroth-order expansion of the transfer equation therefore tells us that

$$I_0(\vec{x}, t, \nu) = B(\nu, T_r)\,. \tag{3.67}$$

Now that we have demonstrated that $S_0(I_0) = 0$, we can further simplify $S_2^{enex}(I_0)$ in equation 3.64 to

$$\begin{aligned} S_2^{enex}(I_0) \;=\; & \oint\Big\{\int_0^\infty d\nu' \int_0^\infty N(p_e, T_m) dp_e \sigma(\nu \to \nu', \Omega \to \Omega') \times \\ & I_0[1 + \frac{c^2}{2h\nu'^3} I_0'] \delta(\nu - \nu') - \\ & \int_0^\infty d\nu' \int_0^\infty N(p_e', T_m) dp_e' \sigma(\nu' \to \nu, \Omega' \to \Omega) \times \\ & \frac{\nu}{\nu'} \frac{d\nu'}{d\nu} I_0'[1 + \frac{c^2}{2h\nu^3} I_0] \delta(\nu - \nu')\Big\} d\Omega'\,. \end{aligned} \tag{3.68}$$

which is the same as $S_0(I_0)$ except that the electron number densities are given at the material temperature while the specific intensities are given at the radiation temperature. Similarly, we can simplify $S_2^{Compton}(I_0)$, equation 3.65, to

$$\begin{aligned} S_2^{Compton}(I_0) \;=\; & \oint\Big\{\int_0^\infty d\nu' \int_0^\infty N(p_e, T_r) dp_e \sigma(\nu \to \nu', \Omega \to \Omega') \times \\ & I_0[1 + \frac{c^2}{2h\nu'^3} I_0'] \delta(\nu - \nu' - (E_e - E_e')/h) - \\ & \int_0^\infty d\nu' \int_0^\infty N(p_e', T_r) dp_e' \sigma(\nu' \to \nu, \Omega' \to \Omega) \times \\ & \frac{\nu}{\nu'} \frac{d\nu'}{d\nu} I_0'[1 + \frac{c^2}{2h\nu^3} I_0] \delta(\nu - \nu' - (E_e - E_e')/h)\Big\} d\Omega' \end{aligned} \tag{3.69}$$

First Order

Writing out the transfer equation expanded to first order gives us

$$(\vec{n} \cdot \vec{\nabla}) I_0 = -\kappa_a I_1 - S_1(I_1) \,. \tag{3.70}$$

If we take the integral over all angles and frequencies of this equation, we can demonstrate that both sides conserve energy. As was already shown in equation 3.35, these integrals annihilate the total elastic scattering operator. The same is true of any order of the elastic scattering operator in a perturbation expansion. We know that $S_1(I_1)$ is elastic because the inelastic scattering terms are all second order as defined by equation 3.30. Taking the angle integral of $-\kappa_a I_1$ gives

$$\oint d\Omega(-\kappa_a I_1) = -\kappa_a \oint d\Omega I_1 = 0 \tag{3.71}$$

because we are assuming that the absorption opacity is isotropic and have set up the perturbation analysis such that all of the angle-integrated intensity is zeroth-order. Now we can write

$$\int_0^\infty d\nu \oint d\Omega \Big(-\kappa_a I_1 - S_1(I_1) \Big) = 0 \tag{3.72}$$

which shows that the first-order collision operator conserves energy. This must be balanced on the left-hand side of the equation by an operator which integrates to zero as well. This is the case, as the angle integral of a unit vector dotted to an isotropic quantity is always zero

$$\oint d\Omega(\vec{n} \cdot \vec{\nabla}) I_0 = 0 \,. \tag{3.73}$$

We are interested in solving for I_1 so we need to invert the collision operator such that we can write

$$I_1 = \mathcal{C}^{-1}[(\vec{n} \cdot \vec{\nabla}) I_0] \,. \tag{3.74}$$

We will propose a solution for I_1 and then demonstrate that it is consistent. In order to get an idea of the functional form of I_1 we proceed with the analysis as Sampson did, neglecting the scattering term. In this case, we can write the first-order

transfer equation as

$$(\vec{n}\cdot\vec{\nabla})B(T_r) = -\kappa_a I_1 \tag{3.75}$$

where we have already substituted in the results from the zeroth-order expansion. This suggests that the functional form of I_1 may be

$$I_1 = \lambda(\vec{n}\cdot\vec{\nabla})B(T_r) \tag{3.76}$$

where λ is some function to be determined that may be frequency and temperature dependent. For this no-scattering case, the form for λ is obviously $\lambda = 1/\kappa_a$. Determining the inverse of the collision operator is now reduced to finding the form of λ.

This is achieved by plugging in the known result for I_0 and the suggested form for I_1 into the equation for $S_1(I_1)$, 3.61. Factoring out a term of $\lambda(\vec{n}\cdot\vec{\nabla})B(T_r)$ leads to the following equation:

$$\begin{aligned} S_1(I_1) \;=\; & \lambda(\vec{n}\cdot\vec{\nabla})B(T_r)\int_0^\infty d\nu \oint d\Omega' \int_0^\infty dp_e N(p_e,T_r)\times \\ & \sigma(\nu\to\nu',\Omega\to\Omega')\Big\{1+\frac{c^2}{2h\nu'^3}B'(T_r)[1-\exp(-\frac{h(\nu-\nu')}{kT_r})]- \\ & \frac{\lambda'(\vec{n'}\cdot\vec{\nabla})B'(T_r)}{\lambda(\vec{n}\cdot\vec{\nabla})B(T_r)}[-\frac{c^2}{2h\nu'^3}B(T_r)+\frac{\nu^3}{\nu'^3}(1+\frac{c^2}{2h\nu^3}B(T_r))\times \\ & \exp(-\frac{h(\nu-\nu')}{kT_r})]\Big\}\delta(\nu-\nu')\,. \end{aligned} \tag{3.77}$$

If we now write the full first-order transfer equation like this

$$(\vec{n}\cdot\vec{\nabla})B(T_r) = -\kappa_a I_1 + S_1(I_1) \tag{3.78}$$

and substitute in equations 3.76 and 3.77, we can cancel out a factor of $\lambda(\vec{n}\cdot\vec{\nabla})B(T_r)$ from each term, leaving us with

$$\lambda^{-1} \;=\; -\kappa_a + \oint d\Omega' \int_0^\infty dp_e N(p_e,T_r)\sigma(\nu\to\nu',\Omega\to\Omega')\times$$

$$\left\{1+\frac{c^2}{2h\nu'^3}B'(T_r)[1-\exp(-\frac{h(\nu-\nu')}{kT_r})]-\frac{\lambda'(\vec{n'}\cdot\vec{\nabla})B'(T_r)}{\lambda(\vec{n}\cdot\vec{\nabla})B(T_r)}\times\right.$$
$$\left.[-\frac{c^2}{2h\nu'^3}B(T_r)+\frac{\nu^3}{\nu'^3}(1+\frac{c^2}{2h\nu^3}B(T_r))\exp(-\frac{h(\nu-\nu')}{kT_r})]\right\}. \quad (3.79)$$

We can solve directly for $(\vec{n}\cdot\vec{\nabla})B(T_r)$ by referring to equation 3.13, which gives us

$$(\vec{n}\cdot\vec{\nabla})B(T_r) = -\frac{2h^2\nu^4\exp(\frac{h\nu}{kT_r})}{c^2kT_r^2[\exp(\frac{h\nu}{kT_r})-1]^2}(\vec{n}\cdot\vec{\nabla})T_r \quad (3.80)$$

$$(\vec{n'}\cdot\vec{\nabla})B'(T_r) = -\frac{2h^2\nu'^4\exp(\frac{h\nu'}{kT_r})}{c^2kT_r^2[\exp(\frac{h\nu'}{kT_r})-1]^2}(\vec{n'}\cdot\vec{\nabla})T_r \quad (3.81)$$

Substituting these values into equation 3.79 allows us to simplify our formulation for λ^{-1} to

$$\lambda^{-1} = -\kappa_a+\oint d\Omega'\int_0^\infty dp_eN(p_e,T_r)\sigma(\nu\rightarrow\nu',\Omega\rightarrow\Omega')\times$$
$$\left\{1-\frac{\exp(\frac{2h\nu'}{kT_r})}{\exp(\frac{2h\nu}{kT_r})}\frac{(\exp(\frac{h\nu}{kT_r})-1)^2}{(\exp(\frac{h\nu'}{kT_r})-1)^2}\frac{\lambda'}{\lambda}\frac{\nu'}{\nu}\frac{(\vec{n'}\cdot\vec{\nabla})T_r}{(\vec{n}\cdot\vec{\nabla})T_r}\right\}. \quad (3.82)$$

As Sampson showed in the appendix of his paper[39], the only part of $(\nu'\vec{n'}\cdot\vec{\nabla}T_r)(\nu\vec{n}\cdot\vec{\nabla}T_r)^{-1}$ that isn't annihilated by the integral over azimuthal angle in $d\Omega'$ and dp_e is $(\nu'\nu^{-1}\cos\theta)$ where θ is the angle between $\vec{n}$ and $\vec{n'}$. Therefore we can make this replacement in equation 3.82 to get

$$\lambda^{-1} = -\kappa_a+\oint d\Omega'\int_0^\infty dp_eN(p_e,T_r)\sigma(\nu\rightarrow\nu',\Omega\rightarrow\Omega')\times$$
$$\left\{1-\frac{\exp(\frac{2h\nu'}{kT_r})}{\exp(\frac{2h\nu}{kT_r})}\frac{(\exp(\frac{h\nu}{kT_r})-1)^2}{(\exp(\frac{h\nu'}{kT_r})-1)^2}\frac{\lambda'}{\lambda}\frac{\nu'}{\nu}\cos\theta\right\}. \quad (3.83)$$

From this point, the form of λ is generally calculated numerically and is a function of the material temperature and dimensionless photon frequency[3, 39]. However, as Sampson stated, the value of λ at a particular T_r and ν is actually the photon mean free path

$$\lambda^{-1}=\kappa_a+\kappa_s=\kappa_t\,. \quad (3.84)$$

We can now rewrite equation 3.76 in its proper form

$$I_1 = \frac{1}{\kappa_t}(\vec{n} \cdot \vec{\nabla})B(T_r) \tag{3.85}$$

which is the correct value for I_1 with the full scattering contribution to the collision operator.

Second Order

To arrive at a diffusion equation form from this perturbation analysis, we must also examine the second-order transfer equation. It is not necessary to solve the equation as it was for zeroth and first order. The integrability condition for the solution of I_2 provides the energy transport equation. This condition annihilates I_2. Thus, while the energy transport equation itself is second-order, the second-order order intensity, I_2, is not required. We can write the second-order terms of the transfer equation expansion in the following form

$$\frac{1}{c}\frac{\partial I_0}{\partial t} + (\vec{n} \cdot \vec{\nabla})I_1 = \kappa_a(B(T_m) - B(T_r)) - \kappa_a I_2 - S_2 \tag{3.86}$$

where S_2 is the linear combination of all the second-order scattering contributions given in equation 3.59. Then, we take angle and frequency integrals of each term.

For the time derivative term, we first integrate over all angles to produce

$$\oint d\Omega \frac{1}{c}\frac{\partial I_0}{\partial t} = \frac{1}{c}\frac{\partial I_0}{\partial t}\oint d\Omega = \frac{4\pi}{c}\frac{\partial I_0}{\partial t}\,. \tag{3.87}$$

Next we substitute in $\mathcal{E}_\nu = 4\pi I_0/c$ to write this in terms of the frequency-dependent radiation energy density

$$\oint d\Omega \frac{1}{c}\frac{\partial I_0}{\partial t} = \frac{\partial \mathcal{E}_\nu}{\partial t}\,. \tag{3.88}$$

Finally, we integrate both sides over all frequencies

$$\int_0^\infty d\nu \oint d\Omega \frac{1}{c}\frac{\partial I_0}{\partial t} = \frac{\partial \mathcal{E}}{\partial t} \tag{3.89}$$

which gives us the time derivative of the frequency-integrated radiation energy density.

Moving on to the $(\vec{n}\cdot\vec{\nabla})I_1$ term, we first substitute in the result of the first-order expansion for I_1 and then integrate over all angles

$$\oint d\Omega(\vec{n}\cdot\vec{\nabla})I_1 = \oint d\Omega(\vec{n}\cdot\vec{\nabla})\lambda(\vec{n}\cdot\vec{\nabla})B(T_r,\nu)\,. \tag{3.90}$$

We assume that λ is isotropic though we do allow for an explicit temperature dependence (λ includes the opacities, after all). This allows us to pull $\vec{\nabla}\lambda$ out of the integral and write

$$\oint d\Omega(\vec{n}\cdot\vec{\nabla})I_1 = \vec{\nabla}\lambda\oint d\Omega\vec{n}\vec{n}\cdot\vec{\nabla}B(T_r,\nu)\,. \tag{3.91}$$

Recall that

$$\oint d\Omega\vec{n}\vec{n} = 4\pi\frac{1}{3}\overline{\overline{\mathcal{I}}} \tag{3.92}$$

where $\overline{\overline{\mathcal{I}}}$ is the identity tensor. Combining the previous two equations allows us to write

$$\oint d\Omega(\vec{n}\cdot\vec{\nabla})I_1 = \vec{\nabla}\frac{\lambda}{3}\vec{\nabla}4\pi B(T_r,\nu) \tag{3.93}$$

or

$$\oint d\Omega(\vec{n}\cdot\vec{\nabla})I_1 = \vec{\nabla}\frac{1}{3\kappa_t}\vec{\nabla}4\pi B(T_r,\nu)\,. \tag{3.94}$$

Now we integrate both sides over all frequencies

$$\int_0^\infty d\nu\oint d\Omega(\vec{n}\cdot\vec{\nabla})I_1 = \frac{4\pi}{3}\int_0^\infty d\nu\vec{\nabla}\frac{1}{\kappa_t}\vec{\nabla}B(T_r,\nu)\,. \tag{3.95}$$

The spatial dependence of the blackbody spectrum is entirely due to the spatial dependence on temperature, so we can make the substitution

$$\vec{\nabla}B(T_r,\nu) = \frac{\partial B(T_r,\nu)}{\partial T_r}\vec{\nabla}T_r \tag{3.96}$$

which allows us to write

$$\int_0^\infty d\nu \oint d\Omega(\vec{n}\cdot\vec{\nabla})I_1 = \frac{4\pi}{3}\vec{\nabla}\int_0^\infty d\nu \frac{1}{\kappa_t}\vec{\nabla}\frac{\partial B(T_r,\nu)}{\partial T_r}\,. \tag{3.97}$$

Now we define a weighted average inverse total opacity, $1/\bar{\kappa_t}$, which makes the following equality possible:

$$\frac{1}{\bar{\kappa_t}}\int_0^\infty d\nu \frac{\partial B(T_r,\nu)}{\partial T_r} = \int_0^\infty d\nu \frac{1}{\kappa_t}\frac{\partial B(T_r,\nu)}{\partial T_r}\,. \tag{3.98}$$

Solving for $\bar{\kappa_t}$ shows that this mean opacity must be

$$\bar{\kappa_R} \equiv \frac{\int_0^\infty d\nu \frac{\partial B(T,\nu)}{\partial T}}{\int_0^\infty d\nu \frac{1}{\kappa_t}\frac{\partial B(T,\nu)}{\partial T}}\,. \tag{3.99}$$

This is the definition of the *Rosseland mean opacity* and will be denoted with a subscript R to differentiate it from other possible weighting schemes for opacity. Notice that the r subscript has been dropped from the temperature — $\bar{\kappa_R}$ is a function of temperature and this general definition will hold for any temperature. We use this weighted mean opacity to rewrite equation 3.95 as

$$\int_0^\infty d\nu \oint d\Omega(\vec{n}\cdot\vec{\nabla})I_1 = \frac{4\pi}{3}\vec{\nabla}\frac{1}{\bar{\kappa_R}}\vec{\nabla}\int_0^\infty d\nu B(T_r,\nu) \tag{3.100}$$

and once again make the substitution for $\mathcal{E}_\nu = 4\pi B(T_r,\nu)/c$

$$\begin{aligned}\int_0^\infty d\nu \oint d\Omega(\vec{n}\cdot\vec{\nabla})I_1 &= \frac{c}{3\bar{\kappa_R}}\vec{\nabla}^2\int_0^\infty d\nu\mathcal{E}_\nu \\ &= \vec{\nabla}\frac{c}{3\bar{\kappa_R}}\vec{\nabla}\mathcal{E}\end{aligned} \tag{3.101}$$

Moving on to the right-hand side of equation 3.86, we can immediately dispense with the elastic scattering terms, since we have already shown in equation 3.35 that the integral over all angles and frequencies of the elastic scattering operator is zero. Therefore the angle and energy integrals of $S_2^{lin}(I_2)$, $S_2^{nl}(I_1)$, and $S_2^{enex}(I_0)$ are all zero.

The angle integrated specific intensity is all contained within I_0. For this reason, combined with the assumption of κ_a isotropy, the $\kappa_a I_2$ term in the second-order transfer equation is also zero when integrated over all angles and frequencies.

Taking the integral over all angles of the split-temperature energy exchange term in equation 3.86 gives

$$\oint d\Omega \kappa_a (B(T_m, \nu) - B(T_r, \nu)) = 4\pi \kappa_a (B(T_m, \nu) - B(T_r, \nu)) \,. \tag{3.102}$$

If we split this result into its two components and just examine the frequency integral of the first term, we can see that we are again faced with pulling a frequency-dependent opacity outside of an integral

$$4\pi \int_0^\infty d\nu \kappa_a B(T_m, \nu) \,. \tag{3.103}$$

We treat this case the same as we did when we derived the Rosseland mean opacity. First we postulate that there is a $\bar{\kappa_a}$ that solves the following equality

$$\bar{\kappa_a} \int_0^\infty d\nu B(T_m, \nu) = \int_0^\infty d\nu \kappa_a B(T_m, \nu) \tag{3.104}$$

and then solve for the mean opacity. This weighting scheme is different from the Rosseland weighting, so we give it a different subscript

$$\bar{\kappa_P} = \frac{\int_0^\infty d\nu \kappa_a B(T, \nu)}{\int_0^\infty d\nu B(T, \nu)} \tag{3.105}$$

where, again, we have dropped the m subscripts as this definition holds for any temperature. This is the definition of the *Planck mean opacity.* We can integrate equation 3.102 over frequency now and proceed in the following way

$$\begin{aligned}
\int_0^\infty d\nu \oint d\Omega \mathcal{X} &= 4\pi \int_0^\infty d\nu \kappa_a B(T_m, \nu) - 4\pi \int_0^\infty d\nu \kappa_a B(T_r, \nu)) \\
&= 4\pi \bar{\kappa_P}(T_m) \int_0^\infty d\nu B(T_m, \nu) - 4\pi \bar{\kappa_P}(T_r) \int_0^\infty d\nu B(T_r, \nu)) \\
&= 4\pi \bar{\kappa_P}(T_m) a T_m^4 - c \bar{\kappa_P}(T_r) \int_0^\infty d\nu \mathcal{E}_\nu
\end{aligned}$$

$$= 4\pi\bar{\kappa_P}(T_m)aT_m^4 - c\bar{\kappa_P}(T_r)\mathcal{E}\,. \tag{3.106}$$

Notice that because the two Planck functions were based on different temperatures, we end up with two separate Planck mean opacities.

We have now evaluated the angle and frequency integral for every term in the second-order transfer equation with the exception of the inelastic Compton scattering term. Combining equations 3.89, 3.101, and 3.106 shows the result:

$$\frac{1}{c}\frac{\partial\mathcal{E}}{\partial t} + \vec{\nabla}\frac{1}{3\bar{\kappa_R}}\vec{\nabla}\mathcal{E} = \frac{4\pi}{c}\bar{\kappa_P}(T_m)aT_m^4 - \bar{\kappa_P}(T_r)\mathcal{E} - \int d\Omega d\nu S_2^{Compton}(I_0) \tag{3.107}$$

where a is the Stefan-Boltzmann constant. This equation is in the form of a diffusion equation for the frequency-integrated radiation energy density, $\mathcal{E}$, and the terms on the right-hand side are source and sink terms due to coupling between the radiation field and the matter.

When we generated the second-order scattering term expansion in equation 3.59, we explicitly put all Compton scattering contributions into $S_2^{Compton}$. Because this is the only term with an energy-conserving delta function that allows for transfer of energy between the photon and the electron, it is the only term to which Compton scattering contributes. Furthermore, as we can see in equation 3.69, there are no surviving terms in $S_2^{Compton}$ that are not due to Compton scattering. Thus, this term contains all of the Compton contributions and nothing else.

Weymann[44] expands on Dreicer[8] to demonstrate that the time development of the radiation energy spectrum for an isotropic photon gas interacting solely through the Compton effect with an electron gas having a Maxwellian distribution is given by

$$\frac{\partial n_\nu}{\partial t} = N_e c\frac{8\pi}{3}\sigma_0^2\frac{2kT_m}{m_e c^2}\left\{\nu(2 + \frac{\nu}{2}\frac{d}{d\nu})[n_\nu(1+n_\nu) + \frac{dn_\nu}{d\nu}]\right\} \tag{3.108}$$

where σ_0 is the Thomson scattering cross section, m_e is the electron rest mass, and N_e is the electron density.

Weymann continues by demonstrating that this can be transformed into a rate

of change of radiation energy density by multiplying equation 3.108 by the photon energy, $h\nu$, and integrating over all momentum space. In the case where $n(\nu)$ is Planckian but is at a temperature $T_r \neq T_m$, this becomes

$$\frac{d\mathcal{E}}{dt} = 4\mathcal{E}N_e c\frac{8\pi\sigma_0^2}{3}\frac{kT_m - kT_r}{m_e c^2}. \tag{3.109}$$

The assumptions made by Weymann, namely that the electron distribution function is Maxwellian and that the radiation distribution function is Planckian at a possibly different temperature than the material, are exactly the same assumptions we made previously in the derivation of equation 3.107. The fact that equation 3.109 represents the changes in radiation energy density as a result of Compton scattering (and only Compton scattering) means that it is exactly equivalent to the Compton source/sink term in equation 3.107.

$$\int d\Omega d\nu S_2^{Compton}(I_0) = 4\mathcal{E}N_e c\frac{8\pi\sigma_0^2}{3}\frac{kT_m - kT_r}{m_e c^2}. \tag{3.110}$$

We can use this equality to finally construct the complete diffusion equation for radiation energy density

$$\frac{1}{c}\frac{\partial\mathcal{E}}{\partial t} + \vec{\nabla}\frac{1}{3\bar{\kappa_R}}\vec{\nabla}\mathcal{E} = \frac{4\pi}{c}\bar{\kappa_P}(T_m)aT_m^4 - \bar{\kappa_P}(T_r)\mathcal{E} - 4\mathcal{E}N_e c\frac{8\pi\sigma_0^2}{3}\frac{kT_m - kT_r}{m_e c^2}. \tag{3.111}$$

3.7 A contrasting derivation of the diffusion equation

The derivation of equation 3.111 given in the previous section is presented for the first time in this thesis. The following is the canonical derivation used by Pomraning[32]. Note that it makes considerably more assumptions and in the end does not produce a Compton scattering correction.

We start by examining the monochromatic radiation momentum equation, equa-

tion 3.11, and how it simplifies in an environment where $\lambda \ll l$. First, divide both sides by $\frac{\kappa_t}{c}$ to get

$$\frac{1}{\kappa_t c}\frac{\partial}{\partial t}\vec{F} + (\frac{c}{\kappa_t})\vec{\nabla}\cdot\mathsf{P} = \frac{1}{\kappa_t}\oint[\eta_t - \kappa_t I]\vec{n}d\Omega \tag{3.112}$$

Next, rewrite the total opacity terms as functions of the mean-free-path via equation 3.4 and expand the integral on the right-hand side

$$\frac{\lambda}{c}\frac{\partial}{\partial t}\vec{F} + (\lambda c)(\vec{\nabla}\cdot\mathsf{P}) = \oint \eta_t \lambda \vec{n}d\Omega - \vec{F} \tag{3.113}$$

If l is the characteristic length for the problem (and therefore a characteristic time of $\frac{l}{c}$), we can see that the flux term on the left-hand side of equation 3.113 is roughly $\frac{\lambda}{l}$ times the flux term on the right-hand side. This means that the left-hand side flux term is insignificant and can be dropped. Next, recall that the integral over angle of an isotropic function against any vector component is zero. In LTE, we make the claim that emission (both thermal and scattering) is isotropic. Therefore, the emissivity integral in equation 3.113 drops out as well. This leaves us with

$$\vec{F} = -(\lambda c)(\vec{\nabla}\cdot\mathsf{P}) \tag{3.114}$$

At the end of chapter 2 it was demonstrated that the pressure tensor reduces to $\frac{1}{3}\mathcal{E}$ when the radiation field is isotropic. Since LTE implies that the field is isotropic we can simplify the $\vec{\nabla}\cdot\mathsf{P}$ term and reduce equation 3.114 to

$$\vec{F}(\vec{x},t,\nu) = -\frac{c}{3\kappa_t}\vec{\nabla}\mathcal{E}(\vec{x},t,\nu) \tag{3.115}$$

Next examine the energy equation, equation 3.10. Substituting in equation 3.115, expanding the integral, and dividing both sides by $c\kappa_t$ reveals

$$\frac{\lambda}{c}\frac{\partial\mathcal{E}}{\partial t} - \frac{\lambda^2}{3}\vec{\nabla}^2\mathcal{E} = \frac{\lambda}{c}\oint \eta_t d\Omega - \mathcal{E} \tag{3.116}$$

In the diffusion limit where $\frac{\lambda}{l} \ll 1$, both terms on the left-hand side become inconse-

quential in comparison to $\mathcal{E}$ and can be dropped. As was stated in section 3.5, if we are to maintain steady-state in LTE, the total emissivity introducing photons into a certain point in phase space must exactly equal the total opacity times the specific intensity. Furthermore, as a consequence of the combination of equations 3.12 and 3.14, we know that the specific intensity is equal to the Planck function, equation 3.13. Using this knowledge, we can rewrite equation 3.116 as

$$\mathcal{E} = \frac{\lambda}{c} \oint \kappa_t B d\Omega \tag{3.117}$$

This, in turn, can be simplified to the following formula for the monochromatic energy density in the diffusion limit:

$$\mathcal{E}(\vec{x}, t, \nu) = \frac{4\pi}{c} B(\nu, T) \tag{3.118}$$

Finally, we can plug equation 3.118 into equation 3.115 to get the simplified form of the radiative flux in the diffusion approximation

$$\vec{F}(\vec{x}, t, \nu) = -\frac{4\pi}{3\kappa_t(\vec{x}, t, \nu)} \vec{\nabla} B(\nu, T) \tag{3.119}$$

This equation represents a tremendous simplification of the flux and is the central assumption of radiative transfer *in the diffusion limit.* It is accurate to high order in materials with very high opacities. However, this derivation did not take material motion into account - it is only correct in the co-moving frame and will break down if there are significant material accelerations. Similarly, the assumption is invalid near material boundaries or any transparent region.

To illustrate that the radiation field does, in fact, exhibit diffusive behavior in the diffusion limit, plug equation 3.119 back into the energy equation (equation 3.10) and rearrange to give

$$\frac{\partial \mathcal{E}}{\partial t} = \frac{c}{3\kappa_t} \vec{\nabla}^2 \mathcal{E} + 4\pi\eta_t - c\kappa_t \mathcal{E} \tag{3.120}$$

which has the same form as the time dependent diffusion equation

$$\frac{\partial f}{\partial t} = \vec{\nabla}^2 f + \mathcal{F} \tag{3.121}$$

where $\mathcal{F}$ is a source-sink term[23].

Chapter 4

Flux Limited Diffusion

The trouble with radiation transport in the diffusion limit is that it can predict a radiative flux that is far too large [13]. To illustrate this, examine the definitions of radiation energy density (equation 2.5) and radiation flux (equation 2.7) re-written in the following way

$$c\mathcal{E} = \oint I d\Omega \tag{4.1}$$

$$\vec{F} = \oint I \vec{n} d\Omega \tag{4.2}$$

From these two equations, we can see that

$$|\vec{F}| \leq c\mathcal{E} \tag{4.3}$$

In fact, the only time we see $|\vec{F}| = c\mathcal{E}$ is in the streaming wave limit where the specific intensity is just a delta function.

Now, compare this result to the diffusion limit approximation of the flux (equation 3.119), which we will alter by plugging in equation 3.118 and re-write as a Fick's law of diffusion

$$\vec{F} = -D\vec{\nabla}(c\mathcal{E}) \tag{4.4}$$

where D is the diffusion coefficient, $D = \frac{1}{3\kappa_t}$. Obviously, for high gradients in the radiation energy density, one can obtain a flux which violates equation 4.3.

The cause of this problem is rooted in the definition of LTE. We assumed during the derivation of the diffusion limit that we were in LTE, and LTE mandates that the gradients in the radiation field be small. The violation of equation 4.3 is simply the mechanism by which the diffusion limit assumption fails if we are, in fact, not in LTE.

Still, we would like to take advantage of the enormous simplification available in the diffusion limit flux approximation despite not always being in strict LTE. The following two sections go over the two principal methods of modifying the diffusion approximation such that it performs in a physical manner when significantly away from thermodynamic equilibrium.

4.1 The Variable Eddington Factor

Numerous methods have been put forth to overcome this difficulty with inaccurate flux [31]. One popular system for approximating radiation transport more accurately than plain diffusion theory is call the variable Eddington tensor approach. To explain this method, we begin with the radiation transfer equation (3.7) and its first two angular moments (equations 3.10 and 3.11), slightly rearranged

$$\frac{1}{c}\frac{\partial I}{\partial t} + \vec{n}\cdot\vec{\nabla}I + \kappa_t I = \kappa_a B + \frac{c}{4\pi}\kappa_s \mathcal{E} \tag{4.5}$$

$$\frac{\partial \mathcal{E}}{\partial t} + \vec{\nabla}\cdot\vec{F} = \kappa_a(4\pi B - c\mathcal{E}) \tag{4.6}$$

$$\frac{1}{c}\frac{\partial \vec{F}}{\partial t} + c\vec{\nabla}\cdot\mathsf{P} + \kappa_t\vec{F} = 0 \tag{4.7}$$

In order to close this series of moment equations, we need to be able to write P in terms of either $\mathcal{E}$ or $\vec{F}$. To do this, we first define the *Eddington tensor*, T, as

$$\mathsf{T} = \frac{\mathsf{P}}{\mathcal{E}} \tag{4.8}$$

and insert it back into equation 4.7

$$\frac{1}{c}\frac{\partial \vec{F}}{\partial t} + c\vec{\nabla}\cdot \mathsf{T}\mathcal{E} + \kappa_t \vec{F} = 0 \tag{4.9}$$

Now we are free to impose an *a priori* form of T as a function of $\mathcal{E}$ and $\vec{F}$. Of course, doing so is strictly an approximation and is not necessarily an accurate one. However, an Eddington tensor can be selected such that it retains the correct properties in the diffusion and wave limits and that is a reasonable interpretation in the middle ground between these limits.

A large number of Eddington tensors have been postulated or derived in the literature, though the bulk of them are of the form

$$\mathsf{T} = \frac{1-\chi}{2}\mathsf{I} + \frac{3\chi - 1}{2}\frac{\vec{f}\vec{f}}{|\vec{f}|^2} \tag{4.10}$$

where I is the identity tensor and $\vec{f}$ is the dimensionless flux, $\vec{f} = \frac{\vec{F}}{c\mathcal{E}}$ [31]. The scalar χ is called the *Eddington factor*. The Eddington factor is usually written as $\chi(f,\omega)$, a function of the magnitude of the dimensionless flux and of the effective single scatter albedo, ω, defined as

$$\omega = \frac{\kappa_a \frac{4\pi}{c} B + \kappa_s \mathcal{E}}{\kappa_t \mathcal{E}} \tag{4.11}$$

If one assumes that the angular distribution is azimuthally symmetric about $\vec{F}$, then χ is given by

$$\chi = \frac{\int_{-1}^{1} I\vec{\mu}\vec{\mu}d\mu}{\int_{-1}^{1} I d\mu} \tag{4.12}$$

where $\mu = \frac{\vec{n}\cdot\vec{f}}{|\vec{f}|}$. If I is indeed isotropic, as it would be in true LTE, then $\chi = \frac{1}{3}$ and the Eddington tensor reduces to

$$\begin{aligned} \mathsf{T} &= \frac{1-\frac{1}{3}}{2}\mathsf{I} \\ &= \frac{1}{3}\mathsf{I} \end{aligned} \tag{4.13}$$

which, in turn, makes the radiation pressure tensor, via equation 4.8,

$$\mathsf{P} = \frac{1}{3}\mathcal{E}\mathsf{I} \tag{4.14}$$

the correct result for an isotropic specific intensity (see equation 2.18) and for the diffusion limit. Since an isotropic specific intensity implies $|\vec{f}| = 0$, any solution for $\chi(0,\omega)$ should equal $\frac{1}{3}$. In the wave limit, where the specific intensity is a delta function, we get values of $|\vec{f}| = 1$ and $\chi = 1$. Therefore any solution for $\chi(1,\omega)$ should equal 1.

The Eddington factor can be looked at as a measure of the anisotropy of the specific intensity. Since the $f = 0$ result represents absolute isotropy and the $f = 1$ result represents absolute anisotropy, we expect the value of χ to vary monotonically between $\frac{1}{3}$ and 1 as f varies between 0 and 1 [31]. As long as this behavior is met by whatever $\chi(f,\omega)$ function is postulated, the resulting system of equations will exhibit the proper qualities in the wave and diffusion limits.

4.2 Flux Limiting

The flux limiting approach to fixing the inaccuracies of the flux as described by Fick's law, equation 4.4, is to modify the diffusion coefficient such that unusually high gradients in the radiation energy density won't violate equation 4.3. To begin, we close the series of moment equations by replacing equation 4.7 with a generic Fick's law of diffusion

$$\vec{F} = -\frac{D}{\kappa_t}\vec{\nabla}(c\mathcal{E}) \tag{4.15}$$

where the dimensionless diffusion coefficient D is chosen such that the resulting equation fully complies with equation 4.3. The functional form of D can be drawn out of thin air or derived in any number of ways, but in general they are functions of a dimensionless gradient

$$X = \frac{|\vec{\nabla}\mathcal{E}|}{\kappa_t\mathcal{E}} \tag{4.16}$$

and the effective single scatter albedo, ω, defined in equation 4.11.

In order to reconstruct the diffusion coefficient from the diffusion limit result in equation 4.4, the value of D in equation 4.15 has to be $\frac{1}{3}$. In the strict diffusion limit, there are effectively no gradients in the radiation field so the quantity X goes to 0. Therefore, when trying to construct a functional form for $D(X, \omega)$, it should always fit the boundary condition of $D(0, \omega) = \frac{1}{3}$.

In the wave limit, the D function needs to account for the possibility of unusually high gradients. The easiest way to handle this is to have D go to 0 as X goes to infinity. Between the diffusion and wave limits, we expect the D function to be a monotonically decreasing function of X.

An ad hoc diffusion coefficient was proposed [20] with the form

$$D = \frac{1}{3\omega + X} \tag{4.17}$$

This formulation was attractive because it reduces to the correct result in the diffusion limit, decreases monotonically with increasing X, and in the wave limit of infinite gradients, it produces the appropriate streaming result

$$|\vec{F}| = c\mathcal{E} \tag{4.18}$$

Outside of having these three desirable properties, the formulation is not based on anything physical.

A more rigorous approach to developing the diffusion coefficient was used by Levermore [17], who derived a flux limited diffusion coefficient and corresponding Eddington factors from the Chapman-Enskog theory of gases. This formulation took the form

$$D = \frac{1}{\mathsf{X}}(\coth(\frac{\mathsf{X}}{\omega}) - \frac{\omega}{\mathsf{X}}) \tag{4.19}$$

where X is a non-positive-definite dimensionless gradient given by

$$\mathsf{X} = -\frac{\vec{\nabla}\mathcal{E}}{\kappa_t \mathcal{E}} \tag{4.20}$$

This diffusion coefficient retained all of the desirable aspects of equation 4.17 but was also based on a physical derivation. Though it is still an approximation, it is considered for these reasons to be a very good one. In addition to the diffusion coefficient, Levermore produced the corresponding Eddington factor that provide the same results

$$\chi = \coth(\frac{X}{\omega})[\coth(\frac{X}{\omega}) - \frac{\omega}{X}] \tag{4.21}$$

Chapter 5

Grey Transport

At this point, let's re-examine the moments of the radiation transfer equation with the diffusion approximation and Levermore flux limiter included. For the time being, the $\vec{x}$, t, $\vec{n}$, and ν subscripts are being omitted.

$$\frac{\partial \mathcal{E}}{\partial t} + \vec{\nabla} \cdot \vec{F} = \oint [\kappa_a B + \frac{c}{4\pi}\kappa_s \mathcal{E} - \kappa_t B] d\Omega \tag{5.1}$$

$$\vec{F} = -\frac{1}{\kappa_t}\frac{1}{\mathsf{X}}(\coth(\frac{\mathsf{X}}{\omega}) - \frac{\omega}{\mathsf{X}})\vec{\nabla}(c\mathcal{E}) \tag{5.2}$$

Equation 5.2 can be simplified by plugging equation 4.20 in for the $\frac{1}{\mathsf{X}}$ term, which gives

$$\vec{F} = c\mathcal{E}(\coth(\frac{\mathsf{X}}{\omega}) - \frac{\omega}{\mathsf{X}}) \tag{5.3}$$

Equations 5.1 and 5.3 are a closed set of equations which can be solved simultaneously to get solutions for the specific intensity, radiation energy density, and radiation flux at all points in phase space. Unfortunately, these quantities as well as the opacities are still functions of seven variables. This means that solving this system of equations is still intractably expensive.

5.1 Opacity Spectrum Complexity

One of the most difficult aspects of solving the radiation transport equations is modeling the photon frequency dependence of the material opacity and the effect this dependency has on other transport quantities like flux. Significant changes in material opacity over small variances in photon energy can lead to remarkably different behavior of photons with very similar frequencies. Attempting to accurately model this behavior requires a detailed description of the opacity vs. energy spectrum for the local material temperature and density. Unfortunately, this spectrum can be quite complex. As an example, consider figure 5-1, which shows a representative plot of the opacity of iron for photons with energies up to 20 keV.

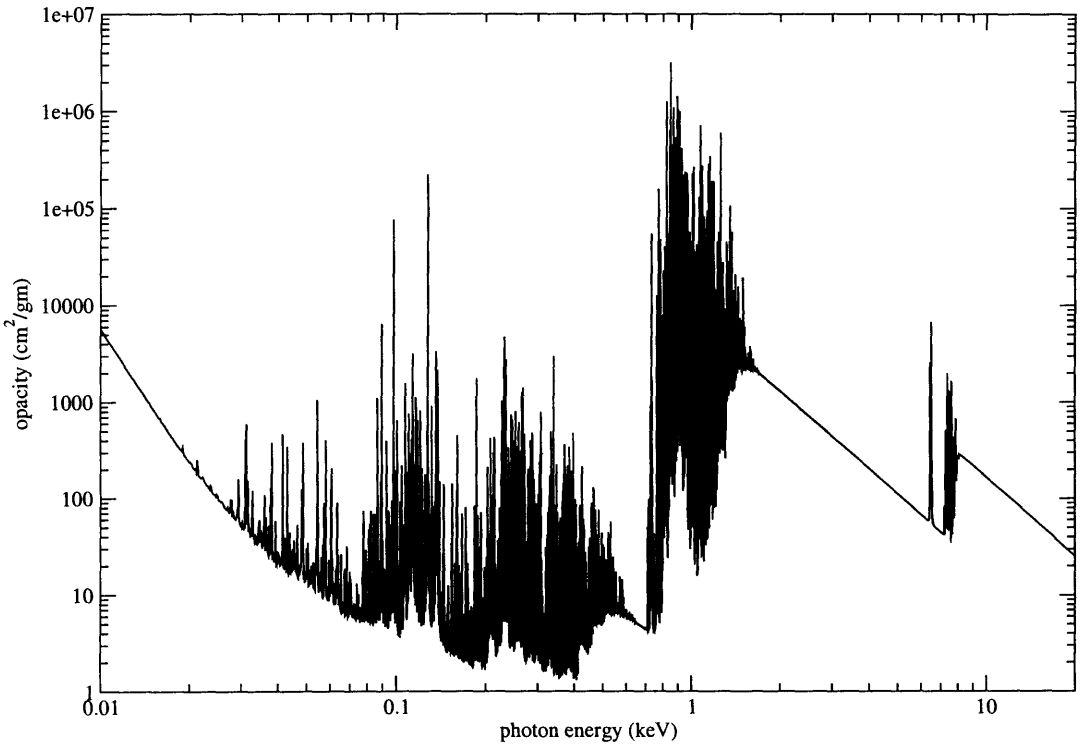

Figure 5-1: Iron opacity for $T = 0.1$ keV and $\rho = 0.0004$ g/cm^3. The opacities in these plots have been normalized by the material density, hence the change in units from the 1/cm seen previously.

Obviously, this particular opacity spectrum contains quite a bit of frequency-dependent structure. The numerous sharp peaks visible in this graph represent bound-

bound absorption interactions between incoming photons and the atomic electrons.

Because of the quantized nature of the electron cloud, each possible energy jump between two permissible energy levels for an atomic electron requires a specific amount of energy. The probability of a jump between two given levels will only be non-zero for photons with the correct energy[1]. Photons with this energy will be subject to the interaction probability caused by bound-free and free-free events plus the bound-bound transition probability. Photons with a slightly different energy will only encounter the bound-free and free-free interaction probability. Thus we see a large peak in opacity at photon energies that correspond to permissible bound-bound electron energy jumps.

The example shown in figure 5-1 was chosen to be a worst-case scenario for iron in the sense that it has a significant amount of spectral noise. In this example, the material is in a density and temperature state that is particularly conducive to bound-bound absorption events for photons in the energy range shown.

Bound-free and free-free absorption events do not give rise to sharp spectral lines because the electrons are not restricted in how much energy they can acquire. To illustrate this point, consider figure 5-2, in which the material temperature is so high that all of the atomic nuclei are fully ionized. In this case, we see a much more well-behaved opacity structure that would be relatively easy to model. However, the assertion that the material is at a temperature of 5 keV means that we are not likely to encounter this environment all that often.

At the other extreme are the low temperatures, where the material is dominated by the near-neutral ion stages. In this case, the electron shells are filled with more electrons than would be present in a hotter atom. This leaves the valence electrons with relatively small ionization potentials and a corresponding restriction to bound-bound transitions with smaller energies. As a result, photons whose energies are

[1]The spectral lines in the opacity are not actually discontinuous—they have a finite width. Apparent photon energies in the material frame are affected by the Doppler shift caused by material motion. The actual line energy can drift slightly as the particle wanders through phase space due to thermal motion. The opacity curves shown here represent an ensemble average over many particles which may have some diffuse distribution in phase space. The spectral lines seen here therefore include some Doppler line-broadening

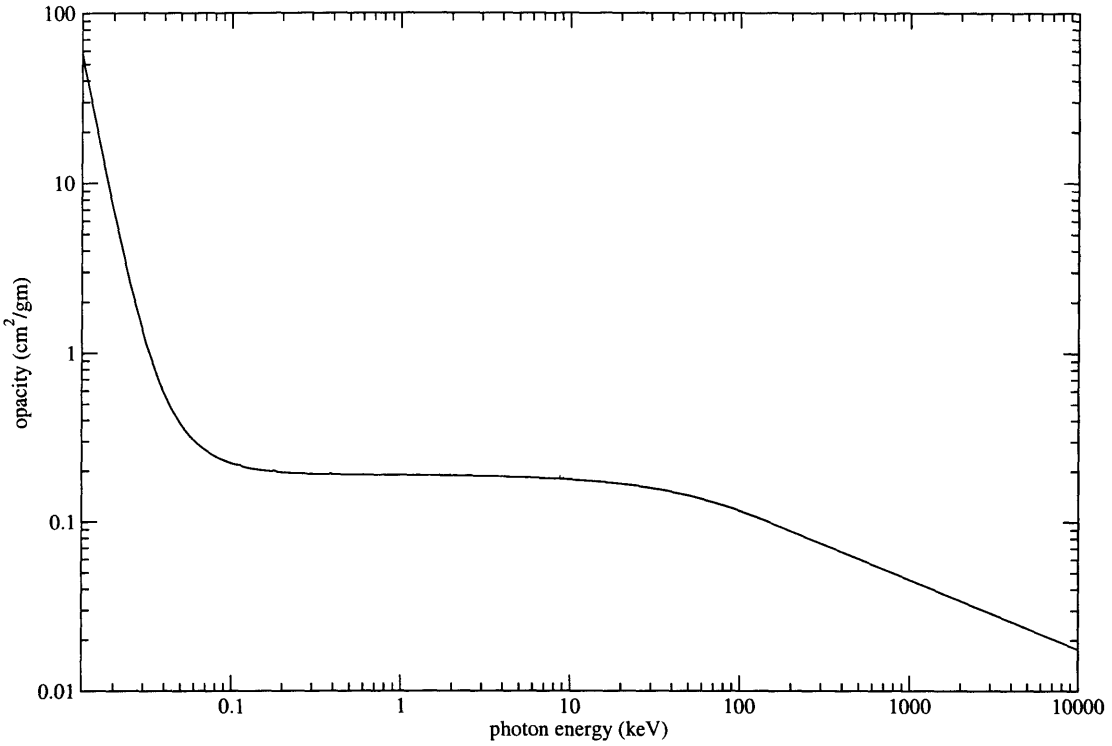

Figure 5-2: Iron opacity for $T = 5$ keV and $\rho = 0.0004$ g/cm^3

above this ionization potential cannot induce a bound-bound interaction.

The end result for the material opacity is a shifting of the lines associated with the valence electrons to lower energies and the absence of higher energy lines beyond a certain photon energy. This absence is due to suppression of bound-bound transitions for the inner-shell electrons as they cannot readily excite to levels that are now occupied by the valence electrons. This is illustrated in figure 5-3, where almost all of the spectral lines occur below the M-edge at 0.02 keV, and only a small number are present just below the L-edge at 0.7 keV.

Clearly, the material opacity varies strongly with material temperature. Furthermore, for material temperatures below full ionization, the opacity structure can be quite complex.

It is important to point out that as the number of bound electrons increases, so does the number of possible bound-bound transitions and thus the complexity of the opacity can increase dramatically. While the number of bound electrons is a function

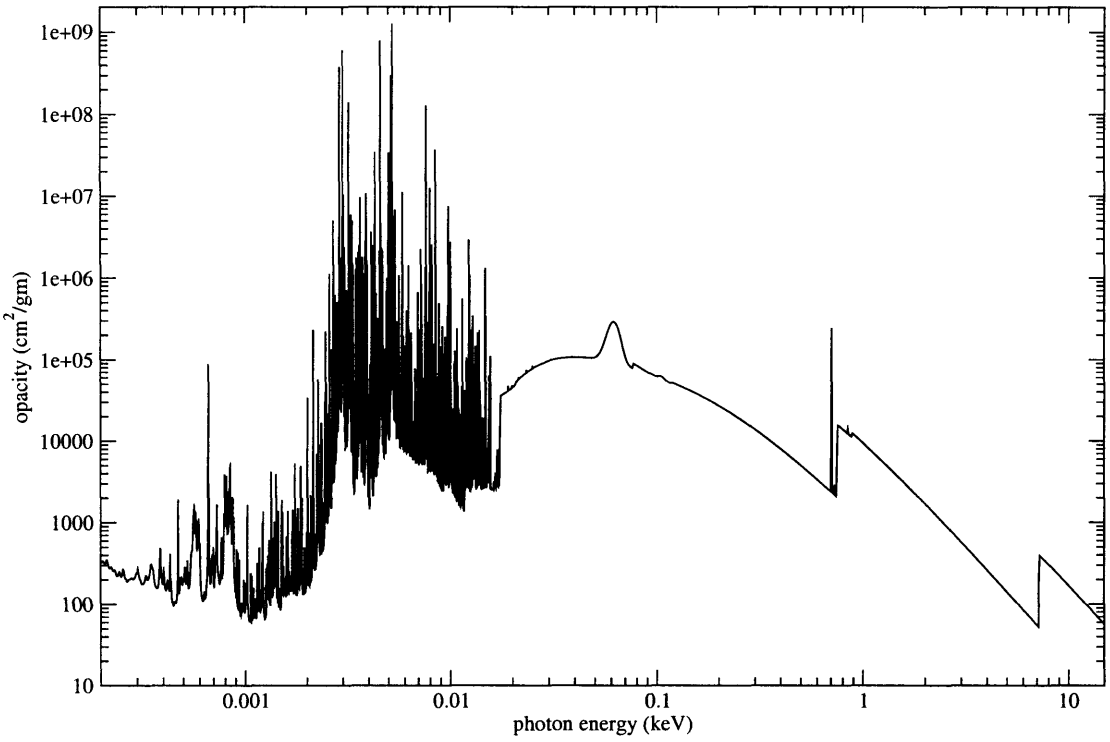

Figure 5-3: Iron opacity for $T = 0.5$ eV and $\rho = 0.0004$ g/cm^3

of the ionization state of the atom, a higher-Z atom will tend to have more bound electrons than a low-Z atom at a given temperature (assuming that the temperature is below the full-ionization energy of the higher-Z atom). The examples given here were all for iron, but opacities with significantly more spectral noise can be obtained from materials in a state with more bound electrons.

5.2 Mean Opacities

When solving realistic radiation transport problems, we are confronted with the issue of modeling the enormously complex opacity vs. photon energy spectrum. As seen in section 5.1, we cannot expect to accurately describe the opacity spectrum with an analytical function. This, combined with the fact that the spectrum is continuously variable, leads to the necessity for approximating the opacity in some way for use in computational codes.

A variety of means for handling this issue have been developed with varying de-

grees of success. Many approximate opacity functions work very well for certain problems and not at all for others. Most offer a tradeoff somewhere between physical accuracy and computational speed.

An example of a highly expensive but very accurate approximation is to collect empirical data for a large number of frequency points on the opacity spectrum. In this way, all of the major features of the opacity function (such as bound-bound absorption lines) are retained. The radiation transport equation is then solved for each of the frequency points. However, for most real-world problems, the opacity function becomes so complex that a prohibitive number of frequency points is required. The number of transport solves needed to complete this approximation rapidly exceeds the computational resources of any realistic computer. Also, there is no guarantee that the problem under examination will involve only those material and energy regimes for which empirical data collection is possible.

At the other end of the tradeoff between speed and accuracy is the mean opacity approximation, also known as the *grey transport* approximation. This is the simplest possible solution to the problem of opacity spectrum complexity. It derives a weighted average opacity that is used for photons of all energies. This derivation is based on the LTE assumption that the photon mean free path, λ, is much less than the scale lengths of the problem, and furthermore that the radiation spectrum is Planckian.

Unfortunately, making such a drastic simplification while still retaining accurate modeling of the physics is not generally possible except in strict thermal equilibrium. Opacity, because it is such a complicated function of frequency, represents a non-linear component in the transport process and cannot be accurately approximated by a constant.

However, there are many possible weighting schemes for taking a mean opacity and some of them produce results with important physical significance. It is possible to construct a mean opacity which, in some regimes, provides a highly accurate approximation to the actual physical behavior in the problem. In the derivation of the diffusion approximation shown in section 3.6, two such opacities appeared. The Rosseland mean opacity was shown to have the correct weighting scheme for the dif-

fusion term while the Planck mean opacity was shown to be correct for the absorption and emission terms.

Chapter 6

Multigroup Transport

In the past, grey transport has been a necessity because of the tremendous savings in computational time it afforded the user. However, as modern computers become more and more efficient, the computational physicist is given greater latitude in compromising between accuracy and speed.

Since the introduction of grey transport, the inaccuracies inherent in the method stuck out like a sore thumb and a higher-fidelity alternative was sought. The natural successor to grey transport is a technique called *multigroup transport.*

The concept of multigroup transport is motivated more by mathematics than any of the specific physical inaccuracies mentioned previously. We start by looking at the material opacity in an abstract sense — as some complicated, continuous, one-dimensional function which we wish to approximate in a discrete way.

If we sample a continuous function, $f(x)$, at n discrete points we can say that the collection of points is a discrete approximation to the continuous function. Intuitively, we can see that the more samples that are taken, the closer the approximation will be to $f(x)$. In order to quantitatively measure the resemblance between the discrete approximation and the continuous function, we can employ a root-mean-square (RMS) analysis of the difference between the two at all points. The smaller the RMS value of the deviation of our approximation from $f(x)$, the better the approximation.

First, we must extend our discrete approximation such that it has a defined value at all points. Instead of defining a representative value at each of the sample points,

we instead break the continuous function up into γ groups with widths Δx_γ, and assign each group a representative value which we hold constant across the width of the group. The discrete approximation therefore consists of a non-continuous stairstep function with γ constant values.

Finding the RMS value of the difference between $f(x)$ and our discrete approximation, $g(x)$, over some interval $x \in [a, b]$ is now a simple integral:

$$RMS = \left[\frac{\int_a^b (f(x) - g(x))^2 dx}{b - a}\right]^{\frac{1}{2}}. \tag{6.1}$$

In the limit that γ goes to infinity, $g(x) = f(x)$ and $RMS = 0$. In general, increasing γ will decrease the RMS difference between $g(x)$ and $f(x)$. There may be some small deviations from this trend due to the discrete nature of $g(x)$ and the possible existence of strong gradients in $f(x)$. However, a significant increase in γ will bring about a reliable decrease in the RMS difference between $g(x)$ and $f(x)$ and thus a better approximation[35].

An extreme example of the deviations from this trend will now be given. Consider the following discontinuous function

$$\begin{aligned} f(x) &= 0 \;\; x \in [0, 1) \\ &= 1 \;\; x \in [1, 2]\,. \end{aligned} \tag{6.2}$$

When we approximate this with a single group over the interval $x \in [0, 2]$, i.e. $\gamma = 1$, we get an average value of $g(x) = 0.5$ everywhere. Equation 6.1 tells us that the RMS difference between $g(x)$ and $f(x)$ is 0.5 over any interval $[a, b]$.

If we now proceed to the $\gamma = 2$ case and assume uniform group spacing over the interval $x \in [0, 2]$, now the group boundary falls exactly on the discontinuity and the value of $g(x)$ is constant within each group. Therefore, $g(x)$ exactly replicates $f(x)$ and we have an RMS difference of 0.

However keeping our group spacing uniform and adding an additional group, $\gamma = 3$, introduces a group that is once again split by the discontinuity. In this case, the

RMS difference of the middle group is still 0.5 while the RMS difference of the outer groups are both 0. The total RMS difference on the interval $x \in [0, 2]$ is 0.0833. This is significantly lower than the single-group RMS value, and yet higher than the two-group RMS value.

It is obvious that any even γ will produce a perfect reproduction of $f(x)$. The odd values of γ will always have one central group that is split in half by the discontinuity and will contribute to some non-zero RMS difference. However, the larger the value of γ, the smaller this contribution will be.

In a contrived case like this one, where we know the location of the discontinuity, it is easy to place a group boundary at the discontinuity and get a perfect approximation with only two groups. In a more general sense, this example should illustrate the fact that the relative position of discontinuities (or near-discontinuities) and the group boundaries can have a significant impact on the accuracy of the approximation.

For an arbitrarily high number of groups, the change in $f(x)$ over Δx is going to be small and the approximation will be a good one. For a modest number of groups, the ability of $g(x)$ to accurately approximate $f(x)$ depends on the the complexity of $f(x)$ and the position of the group boundaries relative to the high gradients in $f(x)$.

The key concept in all of this discussion is that, unless $f(x)$ is a constant, increasing γ will increase the accuracy of $g(x)$ — assuming you have the freedom to place the group boundaries intelligently. This is the motivation for using multigroup radiation transport: more accurately handling the complicated material opacity.

Implementing multigroup transport involves dividing the photon energy spectrum up into some number of contiguous energy groups which span the entire energy range of the problem. For each group, a grey transport solution is calculated. To do this, a set of mean opacities is calculated for each group (see figure 6-1). Because the multigroup mean opacities are not necessarily equal, they retain more information about the actual opacity structure of the material than would a straight grey treatment. Multigroup transport is convenient because it is inherently scalable to fit the computational time available. The multigroup method with only a single group is indistinguishable from grey transport. As the number of groups is increased, the energy

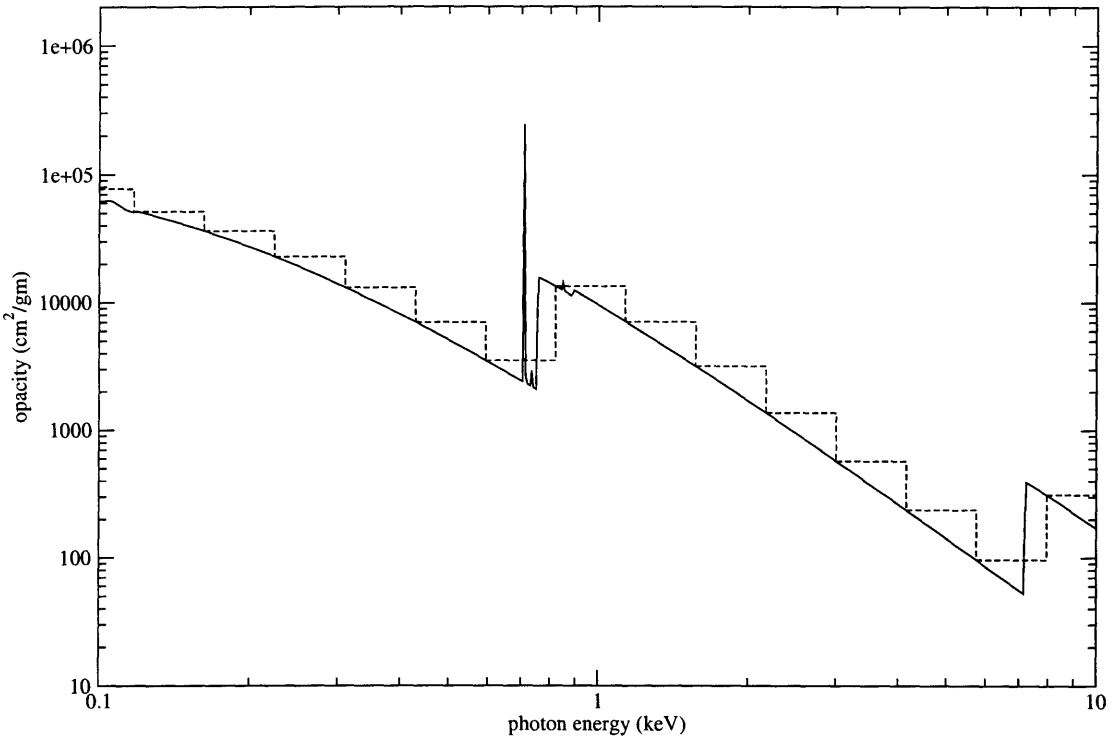

Figure 6-1: A small section of an opacity plot (solid line) superimposed with a corresponding set of multigroup opacities (dashed line). Notice that in this example, the group boundaries are uniformly spaced in log-energy space. Notice also that the discontinuous function represented by the multigroup opacities maintains the general character of the original opacity function, but misses smaller features such as the spike just below 1keV.

width of each group decreases and the amount of actual opacity information that is lost is diminished. As the number of groups goes to infinity, the group width goes to zero, and the mean opacity over a single energy is simply the opacity itself—no information is lost and you effectively have full transport with the actual physical opacity.

However, solving the multigroup diffusion equation requires solving the grey diffusion equation separately for each group. Therefore, the computational needs of the multigroup method scale linearly with the number of groups. In this way, single-group "multigroup" is no slower and no more accurate than grey transport. Likewise, infinite-group transport delivers what is essentially complete accuracy at the expense of eternal computation time. Naturally, the multigroup method is typically exer-

cised somewhere between these two limits. By selecting the number of groups to use, the computational physicist can explicitly choose a point along a continuum of compromise between speed and accuracy.

6.1 Group Boundary Placement

For very large numbers of groups, the specific placement of group boundaries in energy space is irrelevant. When the group widths in energy space are very small, the changes in material opacity over the group width tend to be minuscule. A mean taken over a nearly constant function returns a result that is very accurate at all points. For this reason, the group mean opacities for very thin energy groups (or any group that is relatively constant in opacity) will not lose much information from the actual material opacity.

Unfortunately, computational time is usually at a premium and the impetus is to use as few energy groups as possible in order to reduce the number of transport solves that must be performed. In situations where the number of groups is relatively small, the width of each group in energy space will be large enough that the opacity structure between the group boundaries can be significant. In this case, the averaging process used to generate the group mean opacity could destroy a significant amount of information contained in the opacity spectrum, as described in section 5.2.

However, the multigroup transport method does not impose any rules on the relative widths of the energy groups - they need not be of equal width. In fact, as each group is processed independently, the group boundaries can be placed anywhere in the energy spectrum and there can be a great variance in the group widths. The question arises of whether certain group boundary selections provide benefits over others.

As stated above, a mean opacity taken over a relatively constant region in material opacity is a good approximation to the original. Indeed, any sensible weighting scheme for creating a mean opacity should return the same value if the opacity was constant over the interval in question. For instance, the Planck mean opacity and the Rosseland

mean opacity for a constant-opacity group are equal to each other.

It is only when the opacity within the group is not constant that the averaging process destroys information. Keeping this in mind, an easy way to distribute group boundaries throughout energy space is to place them at significant discontinuities in the opacity spectrum. For instance, at material temperatures below full-ionization, there will always be a significant opacity discontinuity at the K-edge. Also, group boundaries could be placed on either side of spectral lines that the user wished to maintain in the simulation output. In this way, the user can set up the group boundaries by "eye-balling" the problem and defining regions with relatively low opacity gradients. This method is very subjective and can require a lot of setup time on the part of the user.

6.2 Equal Arc Length Projection

A more general method that makes the task of group boundary distribution automatic is called the *equal arc length projection* method. In the context of this technique, *arc length*, L, is defined as the integral of path length, dl, along the opacity function, κ_ν, when plotted versus photon energy.

$$L = \int dl = \int |\frac{d\kappa}{d\nu}| d\nu \tag{6.3}$$

As stated in section 6.1, the less uniform the opacity is within a group, the more information is lost when taking an average opacity over the entire group. Equal arc length projection attempts to minimize these losses by picking energy groups that each contain the same arc length of the opacity function. The measure of opacity arc length divided by the change in energy is referred to as the amount of opacity *structure*, S,

$$S = \frac{\int_{\nu_1}^{\nu_2} |\frac{d\kappa}{d\nu}| d\nu}{(\nu_2 - \nu_1)} \tag{6.4}$$

In order to minimize information loss over the averaging process, we want to ensure that the regions of opacity with high structure are contained in the smallest

groups. Similarly, regions of little structure can have proportionally larger groups. Arranging the group boundaries in this way allows the opacity function arc length within each group to be constant. Therefore, the information loss due to averaging is spread out evenly over all groups and no one group has significantly higher error when compared to the others.

For a known opacity spectrum, this optimal group distribution can be calculated in the following way. First, calculate the total arc length, L_{total}, of the opacity spectrum over the entire energy range.

$$L_{total} = \int |\frac{d\kappa}{d\nu}| d\nu \tag{6.5}$$

Next, calculate the desired arc length per group, L_{group}, by dividing the total arc length by the desired number of groups.

$$L_{group} = \frac{L_{total}}{g} \tag{6.6}$$

Finally, walk along the opacity function calculating the running arc length integral until an integer multiple of the desired arc length per group is reached. At this energy, place a group boundary. In doing so, we have plotted a number of points along the opacity curve that are separated by equal arc lengths. The mathematical description of these points is the set of energies ν_g such that

$$\frac{\int_{\nu_1}^{\nu_g} |\frac{d\kappa}{d\nu}| d\nu}{L_{group}} \tag{6.7}$$

is an integer.

These points are then projected down onto the energy axis and used as the group boundaries. Hence the name of the method, equal arc length projection. Figure 6-2 shows an arbitrarily-chosen iron opacity with the selected group boundaries overlaid as vertical lines.

This method tends to put group boundaries closer together where there are significant gradients in the opacity spectrum and farther apart in relatively constant opacity regions. It will set the boundaries such that the integrated opacity gradient

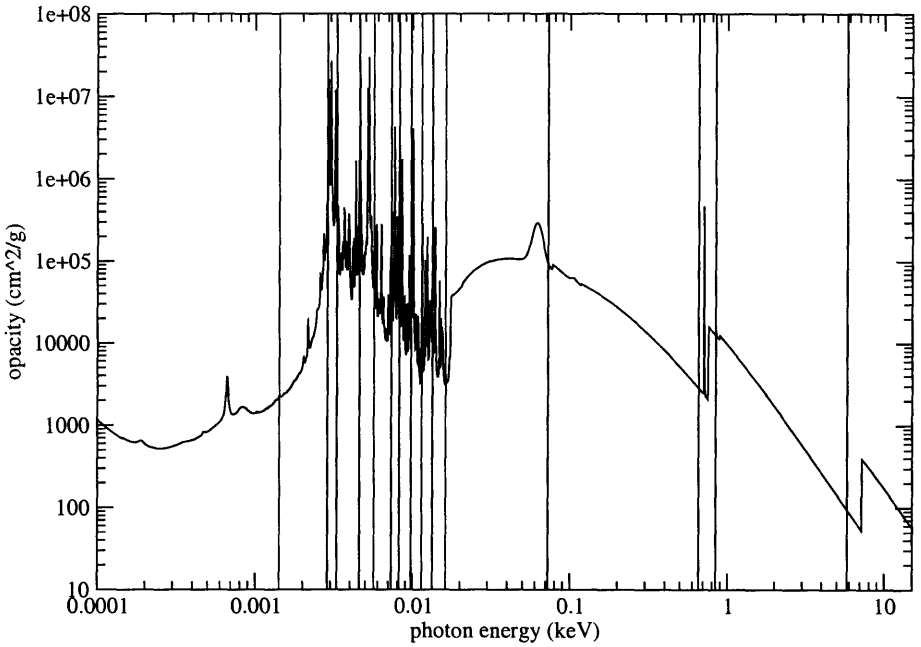

Figure 6-2: An example opacity on which the equal arc length projection method has been performed. The vertical lines indicate the group boundary selections made by the algorithm. Notice that the region of dense bound-bound transition lines has very high group resolution while the relatively smooth regions are characterized by broad energy groups.

across each group is a constant. If the opacity spectrum is very complex and the number of groups is very small, the equal arc length projection method will degrade gracefully and return a constant group width distribution.

6.3 Dealing with changes in the opacity spectrum

As illustrated in section 5.1, the shape of the opacity spectrum can be altered dramatically with changes in material temperature. To a lesser degree, the opacity spectrum is also affected by changes in material density. This leads to problems in distributing group boundaries in energy space because, as the simulation evolves, the temperatures and densities will change and thus so will the opacity spectrum.

A selection of group boundaries chosen manually, by using the equal arc length

projection method (or any other method), on the initial opacity spectrum will be rendered irrelevant if the temperatures and densities in the simulation change significantly in time or space. Rather than using the initial conditions to select the group boundaries, some *a priori* knowledge of how the problem evolves and what temperature/density combination is most important to the simulation can be used. In this way, the user can ensure that in the most important region of phase space, the group boundaries will be rendering the most accurate treatment of the opacity. However, in all other regions of phase space, the opacity treatment could be significantly off.

This difficulty is compounded by the fact that, in most multigroup radiation transport simulation codes - including the RAGE code, the energy group boundaries are specified as part of the input file and do not change during the run. The canonical way of dealing with drastically changing opacities is to stop the simulation mid-run, use the current state as the input for a new simulation, and re-calculate a new energy group structure and the associated multigroup opacities. While this technique works, it is slow, requires a lot of interaction by the user, and the criteria for deciding when to stop the simulation are not well defined.

Chapter 7

Adaption in Energy Space

What is needed to solve the issues described in section 6.3 is an automated system for modifying the group boundaries *during the run of the simulation.* This algorithm must examine the current state of the problem and choose group boundaries that maximize accuracy. This thesis represents the first implementation of such an algorithm. The equal arc length projection method does an excellent job of this when working on a single opacity, but in a real simulation we have to deal with the fact that the material opacities are likely to be different in every cell.

7.1 Accounting for spatial opacity variation

Unfortunately, allowing for different group boundaries in each spatial cell would be infeasible due to issues of computational expense. Separate group structures for each of n cells would entail doing $n * g$ transport solves per time step instead of just g solves, where g is the number of groups. Because the number of cells in a large three-dimensional simulation can be as high as 10^8 or more, it is simply not practical to attempt to treat each cell separately when partitioning energy space.

For this reason, a system had to be devised to deal with a possibly large number of different material opacities present in the problem at any given time. In a sense, a "representative" opacity must be created on which to apply the equal arc length projection technique. The creation of this representative opacity is performed much

like a weighted average of the opacities in each cell. The weighting factors used in this averaging process are derived from a number of physical properties affecting the importance of the opacity in each cell.

The first factor contributing to the weighted opacity average is cell volume. Some simulation environments allow for cells of non-uniform size. This includes adaptive spatial mesh simulation frameworks like the RAGE code. The opacity contribution to the weighted average is normalized to cell volume such that a large cell is weighted more heavily than a small cell. In this way, we can ensure that for a given material distribution, the weighted average will be the same regardless of how the spatial domain is split up into cells - i.e. the representative opacity is mesh-independent.

The next factor that is important to consider when weighting each cell's material opacity is the local radiation energy density. Cells that have a very low radiation energy density possess a relatively small number of photons whereas cells with high radiation energy densities have very large photon populations. Because we are interested in the energy transport caused by the radiation field, it is in our best interest to make a utilitarian decision to place more importance on cells that have higher photon populations. Errors introduced into cells that have little radiation energy density will not result in major energy transport discrepancies. Errors introduced into cells with very large photon counts, on the other hand, could cause arbitrarily bad results. By weighting the representative opacity by the normalized radiation energy density, we ensure that accuracy is preferentially maintained in the most volatile cells.

Another benefit of weighting the average opacity by radiation energy density is that, when in LTE, the cells with the highest radiation energy density will also be those cells with the highest material temperature. In the context of high energy density physics, it is generally the high temperature regions of a simulation that drive the behavior of the simulation as a whole. This is certainly the case in core-collapse supernovae[36] and inertial confinement fusion[2, 18, 19], for example.

7.2 Accounting for temporal opacity variation

The key element of the Adaptive Multigroup Radiation Diffusion algorithm that has not been implemented in the past is the ability to automatically deal with opacities that change in time. The techniques described in section 7.1 need to be performed repeatedly during the run of a simulation in order to handle the evolution of state space in time.

The choice of when to interrupt the flow of the simulation to re-define the energy group structure is made based on two factors. One is a quantitative measurement of how much the state space has changed. The other is a constraint on update frequency based on the computational overhead of the algorithm. If the computational expense of performing the adaption is very high, then we must compromise between accuracy and time by limiting the frequency of updates. If, on the other hand, the overhead is negligible compared with the time taken to perform the multigroup radiation diffusion operation, the adaption could be performed every time step.

An evaluation of the relative overhead of the method will depend strongly on the specific implementation of the radiation package in the simulation framework. For this reason, a discussion of this overhead is left for chapter 8, wherein the specific details of the test implementation of the adaptive multigroup radiation diffusion algorithm are laid out.

For a given radiation package, if it turns out that the algorithm overhead is significant, then a system for measuring the change in state space must be determined. What we are interested in is a quantitative measurement of the change in the "representative opacity" mentioned in section 7.1.

We can simply evaluate the representative opacity at every time step and monitor its variation until the changes are greater than some set threshold value. This evaluation carries with it a certain amount of overhead, but if the evaluation shows that the representative opacity hasn't changed significantly, the remainder of the adaption technique can be skipped at a significant savings in computational expense.

The evaluation of the representative opacity requires examining the current tem-

perature and density in each spatial cell of the simulation. On a parallel computing architecture where the problem is spread out over several processors, this examination will require what is called a "global communication." This involves one processor sending a message to all other processors and then waiting for a reply from each. Global communications are generally considered high overhead and should be kept to a minimum in order to take full advantage of the parallel architecture. Thus, if the simulation is being run on a massively-parallel machine, even the evaluation of the representative opacity could involve significant overhead.

If this were the case, the evaluation could be performed less frequently in order to mitigate the overhead. A user-selectable parameter could be set in the input deck which specified how often to check the representative opacity. This gives the user complete control over the compromise between accuracy and efficiency.

7.3 Mechanics of altering group structure

The actual operation of the adaptive multigroup radiation diffusion algorithm is complicated by the fact that data in a computer are discrete. The evaluation of a representative opacity and the equal arc-length projection method are not undertaken on continuous spectra but instead on some table of data points in computer memory.

7.3.1 Evaluation of discrete opacity data

In order to evaluate a representative opacity as described in section 7.1, the algorithm needs access to some reasonably high-resolution opacity data for a wide range of densities and temperatures for each material in the simulation. This master opacity table can be quite large as it must include at least two of the Rosseland total, absorption, or scattering opacities for each temperature, density, energy, and material combination.

The form of this master table data structure is not specified by the algorithm, but it is convenient to handle these data in the same way as we would the multigroup opacity data in a static multigroup radiation diffusion system. That is to say that we keep a fine-resolution multigroup opacity structure in memory with some arbitrarily

large number of groups and use that to create a coarsened version of the data for use by the radiation package. In this way, we can keep the existing mechanisms for importing and handling multigroup opacity data intact.

The master opacity table can require a large amount of memory. As an example, consider a simulation with two materials, 20 temperature points, 20 densities, and 1,000 energy groups in the high-resolution opacity table. If each opacity value was stored in a 16 byte word, then the table would require 24.4 megabytes of memory. At the time of this writing, a 24MB data structure is rather extreme. However, it is a static data object that need not be communicated between processors and does not change. The memory cost is fixed and depends only on the extent of the master opacity table which can be determined before running a simulation. Also, computer memory has gotten steadily more affordable per bit over the past four decades and what seems like a large amount today will likely be inconsequential on the next generation of computers. Regardless, the composition of the master opacity table can be determined by the user and handled before the simulation is run - again giving control to the user over the compromise between accuracy and memory usage.

Assuming the user has provided the simulation with a high-resolution opacity grid that will fit in the available memory, the algorithm is now free to use these data to do the evaluation of the representative opacity. However, we are using a discrete temperature and density grid in the master opacity table and the (ρ, T) phase space point for a given cell might not lie exactly on a grid point of the master table.

To resolve this problem, we interpolate between the four bounding temperatures and densities in the master table. This interpolation is done for each energy point in the master table to produce a good approximation of the opacity vs. photon energy spectrum for each material in each cell. These spectra can then be combined as described in section 7.1 to form the representative opacity.

7.3.2 Discrete equal arc-length projection

The equal arc-length projection method described in section 6.2 was given in terms of some continuous function. Obviously, because we are now dealing with discrete ener-

gies, we must substitute discrete summations for the integrals and discrete differences for the derivatives in equations 6.5 and 6.7. First, the center energy of each group is calculated in log-energy space. Then, the length of the line segments connecting the group centers is calculated in log-opacity/log-energy space. The total arc length of the opacity function is then just the summation of these line lengths. Equation 6.5 then becomes

$$L_{total} = \sum_{x=0}^{g-1} \sqrt{(E_{x+1} - E_x)^2 + (\kappa_{x+1} - \kappa_x)^2} \tag{7.1}$$

where E_x is the center energy of group x and g is the total number of energy groups in the high-resolution opacity data structure.

Similarly, the handling of equation 6.7 must be altered to account for the discrete nature of the data. In this case, an integer multiple of L_{group} arc-length may fall between two of the energy group boundaries specified in the master opacity table. In this case, the algorithm simply chooses the closest group boundary and selects it to be one of the coarse group structure boundaries.

In theory, accuracy would be improved by interpolating between the two nearest group boundaries in the master table. However, to do so would require a substantial memory allocation as a whole slice of the master opacity table would have to be generated to represent the interpolated data - for each of the coarse group boundaries. In addition, if the master table is generated with sufficiently narrow energy groups, the error will be minimal.

Because of the lack of interpolation between master group boundaries, the possibility of boundary collisions is introduced. If the master table group boundaries are not sufficiently numerous and/or if the requested number of course group boundaries is not significantly less than the number of master table group boundaries, it is possible that two L_{group} increments will fall onto the same master group boundary. These multiply-flagged boundaries must be redistributed elsewhere.

Once the equal arc-length projection method has selected which boundaries are to be used, a quick pass is made to see if any boundary has been selected multiple times. In such cases, the extra boundaries are re-distributed to the nearest available master

table boundary. Such an occurrence is really an indication that the master table has insufficient resolution in that energy band, but due to source data limitations there may be nothing that can be done about it.

7.3.3 Generating the coarse opacity table

Once the evaluation of representative opacity has been undertaken and sufficient changes have occurred to warrant an adaption in energy space, there are several steps that must be taken by the algorithm. As already stated, it first must perform the equal arc-length projection technique described in section 6.2. This provides the algorithm with a new set of coarse group boundaries. These group boundaries are passed off to the multigroup radiation diffusion engine, but are also used to created coarsened versions of all group-dependent quantities in the simulation. Most important of these is the opacity table itself.

The master opacity table cannot be directly used by the radiation engine because it is expecting an opacity data table with just one set of opacities per group. Because the coarsened group structure is what the radiation engine is allowed to see, we must create a corresponding coarsened opacity table to match. This is preferred over the alternative, which is to modify the radiation engine itself to be able to handle an opacity table that does not match the group structure. In the interest of making this algorithm wholly modular, all of the mechanisms for handling the coarsening of group structure and related variables are encapsulated locally such that the radiation package is unaware that anything has changed.

As a result of not interpolating between boundaries in the master opacity table in section 7.3.2, we now have some number of coarse groups each of which is composed of an integer number of high-resolution groups. There are no groups in the master opacity table that are split between two coarse groups. This simplifies the coarsening of the opacities because each group is just the direct combination of some number of master groups.

The technique for combining several Rosseland mean opacities into a single one is not as simple as just averaging. We must ensure that the opacities are combined in

such a way as to maintain the energy-conserving properties of the Rosseland mean. A weighted averaging scheme had to be developed that was "Rosseland friendly."

Re-examining equation 3.99, we can write this in a multigroup formulation where the integrals are not over all energies but only over the energy span of a single group, e.g. group 1 with energies $[\nu_1 ... \nu 2]$

$$\bar{\kappa}_{R:1} = \frac{\int_{\nu_1}^{\nu_2} \frac{\partial B}{\partial T} d\nu}{\int_{\nu_1}^{\nu_2} \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu} \tag{7.2}$$

The Rosseland mean for the next contiguous group with energies $[\nu_2 ... \nu 3]$ would be written

$$\bar{\kappa}_{R:2} = \frac{\int_{\nu_2}^{\nu_3} \frac{\partial B}{\partial T} d\nu}{\int_{\nu_2}^{\nu_3} \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu} \tag{7.3}$$

The question is, how can we take known values for $\bar{\kappa}_{R:1}$ and $\bar{\kappa}_{R:2}$ and use them to create a Rosseland mean opacity for the group spanning energies $[\nu_1 ... \nu 3]$? The resulting desired opacity is defined in the same way

$$\bar{\kappa}_{R:coarse} = \frac{\int_{\nu_1}^{\nu_3} \frac{\partial B}{\partial T} d\nu}{\int_{\nu_1}^{\nu_3} \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu} \tag{7.4}$$

In order to produce this value for $\bar{\kappa}_{R:coarse}$ we can take advantage of two facts. The first is the transitive property of definite integrals which allows us to write

$$\int_{\nu_1}^{\nu_3} \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu = \int_{\nu_1}^{\nu_2} \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu + \int_{\nu_2}^{\nu_3} \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu \tag{7.5}$$

Secondly, we see that the numerators in equations 7.2, 7.3, and 7.4 are not functions of opacity. In fact, all three numerators are functions of known quantities and can be calculated directly. To calculate a definite derivative of $\frac{\partial B}{\partial T}$ we start with the definition of the Planck function as given in equation 3.13 and take its partial derivative with respect to temperature to get

$$\frac{\partial B}{\partial T} = \frac{2h^2\nu^4}{c^2 kT^2} \frac{e^{\frac{h\nu}{kT}}}{(e^{\frac{h\nu}{kT}} - 1)^2} \tag{7.6}$$

Next we make the substitution

$$u = \frac{h\nu}{kT} \tag{7.7}$$

and rewrite equation 7.6 as

$$\frac{\partial B}{\partial T} = \frac{2k^3T^2}{c^2h^2}\frac{u^4e^u}{(e^u-1)^2} \tag{7.8}$$

Plugging this equation into our desired integral over photon frequency we are now looking to solve

$$\int_{\nu_1}^{\nu_2}\frac{\partial B}{\partial T}d\nu = \frac{2k^3T^2}{c^2h^2}\int_{\nu_1}^{\nu_2}\frac{u^4e^u}{(e^u-1)^2}d\nu \tag{7.9}$$

The integral on the right-hand side of this equation has no known analytical solution, but has been formulated such that it has very accurate fits for $u < 0.1$ and $u > 25$. The range $0.1 < u < 25$ can be "calculated" using a numerically-generated lookup table.

This allows us to solve for the numerators in equations 7.2, 7.3, and 7.4. Rearranging 7.2 and 7.3, we can write them in the following form

$$\int_{\nu_1}^{\nu_2}\frac{1}{\kappa_\nu}\frac{\partial B_\nu}{\partial T}d\nu = \frac{\int_{\nu_1}^{\nu_2}\frac{\partial B}{\partial T}d\nu}{\bar{\kappa}_{R:1}} \tag{7.10}$$

$$\int_{\nu_2}^{\nu_3}\frac{1}{\kappa_\nu}\frac{\partial B_\nu}{\partial T}d\nu = \frac{\int_{\nu_2}^{\nu_3}\frac{\partial B}{\partial T}d\nu}{\bar{\kappa}_{R:2}} \tag{7.11}$$

where the right-hand sides of both equations consist entirely of known quantities. This result can be plugged into equation 7.5 to reveal the value for the denominator of equation 7.4. We are left with all known quantities on the right-hand side of 7.4 and thus a known value for the coarse group Rosseland mean opacity.

In this way, we can take any number of contiguous energy groups and combine their mean opacities into a single opacity that maintains the Rosseland weighting scheme. Producing the complete coarse opacity table is a simple matter of undergoing this process for each coarse group defined in the equal arc-length projection step. The only difficult step is solving the integral in equation 7.9 and because we can do this

with a table lookup, the computational overhead for creating the coarse opacity table is extremely low.

7.3.4 Handling group-dependent variables

At this point, the algorithm needs to handle any additional group-dependent variables that the multigroup radiation diffusion engine maintains. How many of these variables there are and what they do will depend on the specific radiation engine in use. In the RAGE code framework used in this thesis, there is only one additional group-dependent variable and that is the multigroup radiation energy density. I will use the treatment of this quantity as an example that can be applied to any group-dependent variable.

The multigroup radiation energy density for group g in cell n is a value which represents how much of the total radiation energy density in cell n is composed of photons whose energies are between the energy boundaries of group g. The values for multigroup radiation energy density are normalized such that the sum of the values for all groups in a given cell will be 1. Thus, to find the actual radiation energy density of photons in group g and cell n, you would take the total radiation energy density for the cell and multiply it by the multigroup radiation energy density for for the group in question.

In theory, the total radiation energy density for a cell should not change as a result of restructuring the multigroup boundaries. Because this value should not change and it happens to be stored separately, we can ignore the total radiation energy density and just work with the weighting factors stored in the multigroup radiation energy array. Our only constraint is that the sum of the values in the array should be 1 for each cell.

Unlike the opacity table, the multigroup radiation energy density values are modified during the execution of the multigroup radiation diffusion engine. Also, the multigroup radiation diffusion engine only works with the coarse group structure. As a result, the coarse multigroup radiation energy density values that are handed off to the multigroup engine will come back modified and the adaptive multigroup radiation

diffusion module must be able to deal with these changes in the weighting factors.

The general solution is to handle the multigroup radiation energy density in much the same way as the opacities - to have a master table of values on a high-resolution group structure, and coarsen this down on the fly. The coarsened group structure has already been determined in previous steps, so we must formulate a method for combining several high-resolution multigroup radiation energy densities into a single coarse one. Unlike the opacity table, where Rosseland behavior had to be maintained, the multigroup radiation energy density values are just simple fractions of the whole and can be added directly.

If group one has 10% of the total radiation energy density and group two as 20%, then a group encompassing both would, by definition, contain 30% of the total radiation energy density. It is quite simple to go through the high-resolution multigroup energy density table and add up the appropriate fine group weights to get the coarse group value. This step requires negligible overhead. The normalization of the group weights should remain intact to within machine precision.

The real effort in handling the multigroup radiation energy density is dealing with the altered values after the multigroup radiation diffusion package has run its course. We have to devise some system of getting these changes reflected in the high-resolution master table. A coarse group which represents the summation of several fine groups may have changed, and this change in value needs to be mapped back onto the high-resolution table. Unfortuantely, because the multigroup radiation package only operates on the coarse energy grid, there is no direct information about how to distribute this change amongst its constituent fine groups.

There are a number of ways to go about distributing these changes. The most simple would be to divide the change in multigroup radiation energy density by the number of fine groups making up the coarse group, then apply this fractional change to each of the fine groups. This method is simple to implement, but it doesn't take advantage of any of the information about the overall pattern of changes across all coarse groups.

A more accurate method would be to examine how the change in one coarse

cell relates to the changes in the neighboring coarse cells[35]. There are numerous interpolation methods that could be used here including polynomial, rational function, and cubic spline interpolation[37, 41]. However, because we are working with a discrete data set and we expect to see some noncontinuous derivatives, the higher-order interpolation functions may be too stiff[13, 43]. Instead, we will use a simple piecewise-linear interpolation

The piecewise-linear interpolation is simple, conservative, and incorporates information about the two neighboring points. For each energy group, the center energy is found (in log-energy space) and these centers are connected by straight lines. For each group, the two lines coming into the center point are examined and the line with the smaller absolute value of slope is chosen and extended through the entire width of the cell. The interpolated value of the change in multigroup radiation energy density at any energy within the group is just the value on the line at the chosen energy.

Because the line is straight and passes through the mean cell value at the center point, the integral under the line stays constant and the interpolation is therefore conservative. To determine the partition of change in multigroup radiation energy density into the high-resolution master table, we simply integrate under the line between the fine-group boundaries and add the result to the old value. In this way, each of the fine group constituents of the coarse group are modified by some fraction of the total change in the coarse group. Furthermore, the sum of the fine group modifications is equal to the total change in the coarse group. Finally, the relative changes in each fine group are related to the relative changes in the neighboring coarse groups and therefore retain more information about that actual behavior of the radiation field.

Once the changes in multigroup radiation energy density have been mapped back into the high-resolution master table, the adaptive multigroup radiation diffusion module is ready to re-create the coarse array once the group boundaries are restructured.

Chapter 8

Implementing Adaptive Multigroup Radiation Diffusion in RAGE

RAGE is a massively parallel continuum radiation hydrodynamics simulation code capable of handling multiple materials in one-, two-, and three-dimensional problems [40]. It was developed by the Thermonuclear Applications group at Los Alamos National Laboratory and by Science Applications International Corporation. The name RAGE is an acronym for Radiation Adaptive Grid Eulerian.

"Adaptive grid" is a reference to RAGE's adaptive mesh refinement (AMR) system which allows for the spatial cells to be split into smaller sub-cells in places where greater spatial accuracy is needed. This adaption in physical space is fundamentally different from the adaption in energy space outlined in this thesis. AMR involves the splitting of mesh cells to form new, smaller cells. This results in a net increase in the number of cells. AMR differs from the adaptive multigroup radiation diffusion algorithm defined here in two principle ways. First, AMR operates in physical space while adaptive multigroup diffusion operates in photon energy space. Second, adaptive multigroup is a "structured adaption" technique, which means that the number of groups is constant and that a given group always has the same two neighboring groups. Structured adaption algorithms tend to be easier to code as the arrangement

of data in memory is more intuitive, but they do not scale well to multiple dimensions. Using a structured algorithm for adaptive multigroup radiation diffusion is appropriate as there is only a single energy dimension upon which to adapt.

"Eulerian" means that the spatial grid is not linked to material motion in the problem and the cell vertices stay fixed in space. Hydrodynamics causes material to flow into and out of cells as the problem evolves. This is contrasted with a *"Lagrangian"* code, in which the cell vertices are linked with a a specific piece of material and are allowed to flow with the hydrodynamics. The benefit of a Lagrangian code is that a given cell always contains the same material, making conservation of mass very simple. However, because the cell vertices move with the material, the cells can get twisted and stretched into long aspect ratio morphologies that are difficult to handle numerically. The cells in an Eulerian simulation are always the same shape (though AMR may cause different scales of cells to arise) so the numerics are somewhat simpler. However, an Eulerian code must be very careful about accounting for mass transfer from one cell to the next and the associated change in state variables.

The RAGE code framework was designed to be platform-independent. Current versions will compile and run under MacOS, Cygwin, IRIX64, AIX, Linux, and TRU64[1] The research outlined in this thesis was performed by the author while working as a graduate research assistant with the Thermonuclear Applications group at Los Alamos National Laboratory.

8.1 Multigroup Radiation Diffusion in RAGE

The specific implementation of multigroup radiation diffusion in RAGE is based on the work of A. M. Winslow[45] and attempts to solve the following time-differenced equations:

$$\mathcal{E}_g - \mathcal{E}_{0g} + \Delta t \nabla \cdot \bar{F}_g = -c\Delta t \kappa_{a,g}(\mathcal{E}_g - b_g a T^4) \tag{8.1}$$

$$\bar{F}_g = -\frac{c}{3\kappa_{t,g}} \nabla \mathcal{E}_g \tag{8.2}$$

[1]RAGE is export controlled and cannot be used off-site without the express permission of LANL's BUS-4 office.

$$\rho C_V(T - T_0) = \sum_{g=1}^{G} c\Delta t \kappa_{a,g}(\mathcal{E}_g - b_g a T^4) \tag{8.3}$$

where g is the group index and G is the total number of groups ($g = 1...G$). A subscript 0 indicates the initial value at the beginning of a timestep. b_g is the Planck weight for group g normalized such that $\sum_{g=1}^{G} b_g = 1$.

Equations 8.1 and 8.2 are the the radiation energy equation and flux description, as derived in chapters 3 and 4. Equation 8.3 is the material energy equation, which relates the change in material temperature to the sum of all photon absorption from all groups.

This set of equations is handled by a method called Multigroup Multi-Frequency-Grey (MGMFG) which attempts to deal with the coupling between equations 8.1 and 8.3 with a two-step process. First, the radiation energy equation is solved separately for each group using the initial value, T_0, for the material temperature and for evaluating the material opacities. By using this known temperature, the frequency groups are decoupled in the radiation energy equation. This pass estimates the spectral distribution of the time-advanced radiation energy density, i.e. it updates the multigroup radiation energy density. This first pass is referred to as the multifrequency pass.

Next, two average opacities are calculated from the results of the multifrequency pass:

$$\kappa_{\mathcal{E}} = \frac{\sum_{g=1}^{G} \kappa_{a,g}\mathcal{E}_g}{\sum_{g=1}^{G} \mathcal{E}_\}} \tag{8.4}$$

$$\kappa_P = \sum_{g=1}^{G} \kappa_{a,g} b_g \tag{8.5}$$

Using these two definitions we can re-write the material energy equation (8.3) as

$$\rho C_V(T - T_0) = c\Delta t(\kappa_{\mathcal{E}}\mathcal{E} - \kappa_P a T^4) \tag{8.6}$$

Next we formulate the equations needed to solve the grey pass, whose goal is to calculate the time-advanced values for the total radiation energy density and the material temperature. First, we sum the radiation energy equation (8.1) over all

groups to get

$$\mathcal{E} - \mathcal{E}_0 + \Delta t \sum_{g=1}^{G} \nabla \cdot \bar{F}_g = -c\Delta t(\kappa_{\mathcal{E}}\mathcal{E} - \kappa_P a T^4) \tag{8.7}$$

Dealing with the flux term in 8.7 requires that we use spatially differenced equations. By integrating 8.1 over cell volume in cell l and then employing the divergence theorem in the flux term, we get

$$V_l(\mathcal{E}_{g,l} - \mathcal{E}_{0,g,l}) + \sum_k \frac{c\Delta t A_{l,k}}{3\Delta\tau_{g,l,k}}(\mathcal{E}_{g,l} - \mathcal{E}_{g,k}) = -c\Delta t V_l \kappa_{a,g,l}(\mathcal{E}_{g,l} - b_{g,l} a T^4) \tag{8.8}$$

where V_l is the volume of cell l, k is the set of all neighboring cells, $A_{l,k}$ is the area of the interface between cells l and k, and $\tau_{g,l,k}$ is the optical path length for group g between cells l and k. This allows us to write the equivalent spatially differenced version of equation 8.7:

$$V_l(\mathcal{E}_l - \mathcal{E}_{0,l}) + \sum_k (\langle a_{l,k}\rangle_l \mathcal{E}_l - \langle a_{l,k}\rangle_k \mathcal{E}_k) = -c\Delta t V_l(\kappa_{\mathcal{E},l}\mathcal{E}_l - \kappa_{P,l} b_{g,l} a T^4) \tag{8.9}$$

where

$$\langle a_{l,k}\rangle_l = \sum_{g=1}^{G} \left(\frac{\mathcal{E}_{g,l} c\Delta t A_{l,k}}{\mathcal{E}_l 3\Delta\tau_{g,l,k}}\right) \tag{8.10}$$

and

$$\langle a_{l,k}\rangle_k = \sum_{g=1}^{G} \left(\frac{\mathcal{E}_{g,k} c\Delta t A_{l,k}}{\mathcal{E}_k 3\Delta\tau_{g,l,k}}\right) \tag{8.11}$$

can be calculated before the grey pass using the results from the multifrequency pass for the radiation energy densities. The averages $\langle a_{l,k}\rangle_l$ and $\langle a_{l,k}\rangle_k$ are defined as the off-diagonal elements of the diffusion matrix D.

Now, turning to the material energy equation formulation for the grey pass, we can integrate equation 8.6 over cell volume to determine the time-advanced temperature:

$$(\rho C_V)_l(T_l - T_{0,l}) = c\Delta t(\kappa_{\mathcal{E},l}\mathcal{E}_l - \kappa_{P,l} a T_l^4) \tag{8.12}$$

This equation can then be added to equation 8.9 to give the following equation which

can be used to determine the time-advanced total radiation energy density, $\mathcal{E}_l$:

$$V_l(\mathcal{E}_l - \mathcal{E}_{0,l}) + \sum_k (\langle a_{l,k} \rangle_l \mathcal{E}_l - \langle a_{l,k} \rangle_k \mathcal{E}_k) = -(\rho C_V)_l (T_l - T_{0,l}) \tag{8.13}$$

The preceding two equations are solved during what is referred to as the grey pass. They are roughly equivalent to the grey diffusion equations described in chapter 3 except that now the diffusion matrix D is non-symmetric and the absorption coefficients $\kappa_\mathcal{E}$ and κ_P are not identical.

Additional details as to the specific mechanics of solving both the multifrequency and grey passes of the MGMFG method in RAGE can be found in Tom Betlach's unpublished work, *"Multifrequency-gray Method for Multigroup Radiation diffusion"*[1].

8.2 Opacities in RAGE and the TOPS code

Equations 8.1, 8.2, and 8.3 all make use of multigroup material opacity coefficients. In order to undergo the MGMFG method, the multigroup radiation diffusion package must have access to the a large table of these opacities.

RAGE imports a multigroup opacity table from an external file at runtime. This file has pre-defined energy group boundaries and if the boundaries in the opacity file do not match the boundaries defined in the simulation input deck, the code will throw an exception and halt. The external opacity file is generated using a separate code called TOPS.

TOPS is maintained by the Diagnostics Methods group at LANL. It allows the user to manipulate the monochromatic opacity information found in the OPLIB opacity databases maintained by the Atomic and Optical Theory group at LANL. The OPLIB databases are enormous collections of data and TOPS acts as a filter to create more reasonably-sized tables with just the information the user needs.

For use with RAGE, TOPS is generally employed to calculate Rosseland mean opacities for user-specified group boundaries and materials or material mixtures. The user is allowed to specify what material temperatures, densities and photon energies

are to appear in the output table in addition to the materials. The entire opacity table created by TOPS must be stored in memory by RAGE, so the user is given the ability to tailor the available data to suit the capabilities of the computer as well as the requirements of the simulation.

For use with the adaptive multigroup radiation diffusion algorithm, the TOPS output defines the high-resolution group structure and associated material opacities. The memory usage of this large table will determine the maximum available energy resolution that can be used when creating the master opacity table in TOPS. The opacity table made available to the multigroup radiation diffusion engine is dynamically generated, relieving the constraint that the input deck must match the TOPS output's energy group boundary structure. Instead, the input deck now only specifies the desired number of groups in the coarsened table and the adaptive multigroup module takes care of the rest, using whatever groups are available in the TOPS output.

In order to fool the radiation diffusion engine into believing that the dynamically-generated opacity table is actually the output from TOPS, it was necessary to maintain the same data structures and formatting. This means that the adaptive multigroup routine must also keep track of the materials, densities, and material temperature information found in the TOPS output and incorporate this into the dynamically generated table.

8.3 Evaluating the overhead of the Adaptive Multigroup method

In section 7.2 the point was made that a decision must be made on how often to adapt in energy space based on the overhead involved in doing so. If the overhead in regrouping is not negligible, then some quantitative measurement of the need to do so must be made. This section analyses the computational overhead of the adaptive multigroup method as implemented in the RAGE code framework. For the purposes

of this investigation we are only interested in the recurring overhead involved with routines that are run every time the adaption in energy space occurs. Initialization routines and one-time costs are ignored.

Each iteration of the adaptive multigroup routine is composed of five principal operations. These are the evaluation of the representative opacity, the determination of new group structure via the equal arc length projection method, building the coarsened opacity table, building the coarsened multigroup radiation energy density table, and remapping the changes in multigroup radiation energy density back into the master table. Each of these steps will be analyzed for its computational cost and the total will be compared against the computational cost of the multigroup radiation solve itself.

8.3.1 Order of magnitude computational cost: Adaptive Multigroup

In order to compare the computational expense of these code blocks, we will define the following parameters: F is the number of energy groups in the high-resolution master opacity table, G is the number of energy groups in the coarsened table, M is the number of materials in the problem, T is the number of temperature points in the opacity tables, R is the number of density points in the opacity tables, and N is the number of cells in the problem. For most simulations the following inequality should hold:

$$N >> F >> G \approx T \approx R >> M \tag{8.14}$$

Evaluating the representative opacity involves looking at the material opacity for each material in every cell of the problem. For each material (M) and cell (N), the algorithm examines the temperature and density of the cell and finds the bounding temperature and density points in the master opacity table via binary decomposition search ($log_2(T+R)$) and then loops over each high-resolution energy group (F) and interpolates the opacity. The result of this interpolation is then incorporated into the representative opacity data structure. This entire computation's computational cost

is $\mathcal{O}(N*(F+log_2(T+R))*M)$. However, $log_2(T+R)$ is going to be inconsequential when compared to F for most simulations, so we can simplify this to $\mathcal{O}(N*F*M)$.

The equal arc-length projection method is comparatively simple to execute as it operates only on a single data structure that was calculated during the previous step. It is composed of a series of loops over all high-resolution groups. Once to calculate the group center energies, once to calculate the derivatives of each segment, once to calculate the arc length of each derivative segment, once to walk through the function and flag the desired group boundaries, and once to deal with any multiply-flagged boundaries. Six loops over high-resolution groups in all, for a total computational cost of $\mathcal{O}(6F)$.

Building the coarsened opacity table involves a separate calculation for each possible combination of material, temperature, density, and coarse group. Once this is done, the population of the data structure itself requires a loop over all materials surrounding two loops over coarse groups. The overall computational cost is therefore $\mathcal{O}(G*T*R*M+2*G*M)$. However, $T*R$ is very likely to by much larger than 2 so we can simplify the cost to $\mathcal{O}(G*T*R*M)$.

Building the coarsened multigroup radiation energy density table involves several steps. First, we initialize a set of coarse group weights for each cell $(N*G)$. Next we renormalize the master multigroup radiation energy density table to make sure the weights sum to 1 $(N*2F)$. Then we loop over all high-resolution groups and sum up the weights for each coarse group in every cell $(N*F)$. Finally, we take the calculated weights and generate the coarse multigroup radiation energy densities for each coarse group and cell $(N*G)$. These values are then renormalized to ensure that the sum of all multigroup radiation energy densities are equal to the grey value $(N*2G)$. The total cost of operation for this routine is therefore $\mathcal{O}(N*(3F+4G)))$ but because $F>>G$ we can simplify this to $\mathcal{O}(3*N*F)$.

The final code segment to be considered is the routine for remapping changes in the coarse multigroup radiation energy density back onto the high-resolution table. This involves looking at every cell and performing a piecewise-linear interpolation on the multigroup radiation energy densities for each. This interpolation consists of a

loop over all coarse energy groups to determine the derivative of multigroup radiation energy density there, then again to distribute the necessary values to the constituent fine energy groups. The total cost of this operation is $\mathcal{O}(2 * N * G)$

Now that we have analyzed the order of magnitude cost of operation for each segment of the adaptive multigroup radiation diffusion package, we can combine them to get an idea of the total cost. In this case, each of the above outlined segments is only run once per timestep so the total cost is simply the sum of the individual parts.

$$\mathcal{O}(N * F * M + 6 * F + G * T * R * M + 3 * N * F + 2 * N * G) \tag{8.15}$$

Re-arranging this equation by extracting a factor of N from every term in which it is found gives:

$$\mathcal{O}(N * (F * M + 3 * F + 2 * G) + 6 * F + G * T * R * M) \tag{8.16}$$

The $2 * G$ term can be dropped as it is insignificant when compared to $3 * F$.

$$\mathcal{O}(N * F * (M + 3) + 6 * F + G * T * R * M) \tag{8.17}$$

Furthermore, 6 can obviously be ignored in comparison to $N * (M + 3)$ so we can drop the $6 * F$ term as well.

$$\mathcal{O}(N * F * (M + 3) + G * T * R * M) \tag{8.18}$$

In most cases, $G * T * R$ is going to be considerably smaller than $N * F$, leading to a final order of operation of approximately

$$\mathcal{O}(N * F * M) \tag{8.19}$$

8.3.2 Order of magnitude computational cost: Multigroup Diffusion

What we are interested in is the the relative cost of the adaptive multigroup algorithm in comparison to its static counterpart. A similar analysis to the previous section can be performed for the standard multigroup radiation diffusion routine. This is the routine that is used to do static multigroup radiation diffusion and is also called when adaptive multigroup is enabled, between the creation of the coarsened multigroup radiation energy density table and the remapping of the changes in that table back up to the master table.

However, since this routine encompasses thousands of lines of code and is not the focus of this document, I will forgo a routine-by-routine analysis of the multigroup radiation diffusion package in RAGE. The bulk of the computational time taken by this package is spent in three calls to the symmetric matrix solver which consists of a matrix-vector multiply and one scalar product per iteration of the solver[38]. However, each solve is tested for convergence and if some convergence criteria isn't met then the matrix solver is called again. In practice, many iterations of the solver are called each timestep. Each iteration is roughly $\mathcal{O}(N)$ plus a significant amount of interprocessor communication time.

The multigroup radiation diffusion engine in RAGE calls the matrix solver once per coarse group during the multigroup pass, then again in the grey pass. The grey pass is iterated to convergence on its own in addition to the overall iteration of the radiation transport process. The order of computational expense for the multigroup radiation diffusion package in RAGE is approximately

$$\mathcal{O}(N * I * (G + J)) \tag{8.20}$$

where I and J are the number of convergence iterations that are performed. These two values change from timestep to timestep but are generally greater than 1. The factor of $I * (G + J)$ is comparable to $F * M$ in equation 8.19, which suggests that the adaptive multigroup overhead is comparable to the amount of time the multigroup

radiation diffusion package requires. However, these order-of-magnitude calculations do not include time spent on interprocessor communication. The adaptive multigroup interprocessor communication consists of a single global collect, whereas the multigroup radiation diffusion package requires communication between each processor and all processors that contain adjacent computational cells. The cost of this neighbor communication scales poorly as the number of physical dimensions and/or processors is increased.

For this reason, it was decided that the adaptive multigroup routines could be run every timestep without a significant loss in performance for RAGE. As a result, no effort was spent on developing a quantitative measurement of the change in representative opacity as discussed in section 7.2.

8.4 Splicing the Adaptive Multigroup module into RAGE

The code segments that make up the adaptive multigroup radiation diffusion module can be separated into three functional blocks. The first is a set of initialization routines which handle setting up the master opacity and multigroup radiation energy density tables. The second block is intended to be run before the radiation solver is engaged and is responsible for dynamically generating the coarsened opacity and multigroup radiation energy density tables. The final code block handles the changes in multigroup radiation energy density after the radiation solver has run and maps those changes back into the master table.

Fortunately, the RAGE code framework was built with modularity in mind. There are numerous points during the execution of any simulation during which RAGE checks a list of hooks to see if any exterior modules wish to run a routine. The adaptive multigroup initialization code block, for instance, is called from a routine that exists specifically to allow external modules to do some initial setup before the timestep cycling begins. The coarsening routines are called during the radiation handler but

just before the multigroup radiation diffusion package's hook is engaged. In this way, the coarsened opacities and radiation energy densities are created with the most current information and used immediately by the radiation package. The multigroup radiation energy density cleanup routine is called by a hook list that engages after all of the physics packages have been executed. In this way, any external modules that affect the multigroup radiation energy density will have their changes incorporated back into the master table correctly.

This modular design and integrated code hook mechanism makes adding the adaptive multigroup radiation diffusion package extremely easy. The entry points for the module are all called from a single hooks code module in RAGE, and the necessary changes to the hook module totaled less than ten lines of FORTRAN. All of the active code for performing the adaptive multigroup algorithm is contained in its own module file.

There were only two additional changes that had to be made to RAGE beyond the inclusion of the adaptive multigroup module source and modifying the hooks. The first was to the multigroup radiation diffusion package. Because the standard multigroup routines assume that the opacity table is constant, they did a few lookups of information from the table during initialization that were then held constant. The content of the opacity table is actually changing frequently, so the routine was modified to look up these values every time they were needed. The overhead for this change was negligible and it had no effect on simulations run without the adaptive multigroup routines enabled.

The other change was to the routine that read information from the TOPS output file. Again, this routine was assuming some of the data in the opacity table to be held constant throughout the simulation. A simple switch was put in place that led to dynamic behavior if the adaptive multigroup routines were active. This involves no overhead or effect on the outcome of simulations without the adaptive multigroup radiation diffusion package enabled.

Chapter 9

Performance Comparisons

9.1 A Figure of Merit

The adaptive multigroup diffusion technique outlined in this document was designed to be as modular as possible. Though it is integrated into the RAGE code for the examples given in this chapter, the technique could just as easily be used for any transport code that involves multigroup discretization of energy space. For instance, it could be used in concert with an implicit Monte Carlo system or even multigroup neutron transport.

However, the specific implementation of adaptive multigroup diffusion places constraints on the capabilities of the system. As described previously, the radiation diffusion package in RAGE has some inherent assumptions which limit its physical accuracy, e.g. the diffusion approximation. For this reason, and because we would like to have an objective measurement of the performance of the technique, the following system was devised to measure the relative merit of using the adaptive multigroup diffusion technique. This method is internally consistent and is independent of the physical basis for the host transport package.

As described in chapter 6, the central idea behind multigroup transport is that it allows for an adjustable-accuracy approximation of the material opacity and photon energy spectrum. In a multigroup radiation diffusion simulation environment like RAGE, we can say that having a very large number of groups will give an inher-

ently better answer than a very small number of groups. This can be done simply on the grounds that the finer group structure includes more information about the monochromatic photon spectrum that would otherwise be lost in the averaging process.

If we further assume that the improvement in accuracy for a small number of groups brought about by enabling adaptive multigroup diffusion is small compared to the improvement brought about by dramatically increasing the number of groups, then we have an internally consistent metric of merit. For a given test problem, a reference run can be made with a very large number of groups. This is then compared against two test runs with and without adaptive multigroup enabled.

If the parameter under study is closer to the reference point with adaption enabled, then there has been an improvement in accuracy. In order to keep our distance from any particular test problem or transport implementation, we can formulate an internally consistent quantitative measurement of the accuracy improvement, as well. We do this by measuring how many groups must be added to the static multigroup test problem in order to match the accuracy of the adaptive multigroup test problem.

If a 10-group adaptive multigroup test problem matches the output for a 12-group static multigroup implementation of the same problem, then we can say that there is a 16.7% reduction in group count allowed for a given accuracy threshold. If we were to ignore overhead, this reduction in group count would be a direct measurement in the reduction in computational time required for the problem. The amount by which overhead eats into this time savings is problem- and platform-specific and will only be discussed for the RAGE implementation and specific test problems found in this document.

9.2 A Simple Test Problem: The Marshak Wave

One of the most simple test problems for radiation transport simulations is the Marshak Wave, first described by R.E. Marshak in 1958 [21]. In this problem, an energy source is introduced into a cold absorber and radiative processes are assumed to

be the only mechanism for energy transport. The result is a propagating thermal wave that diffuses through the material in a way that is very accurately described by equilibrium diffusion theory.

The absorbing matter is assumed to be homogeneous with a constant heat capacity, c_v, such that the material energy density at any point is given by

$$\mathcal{E}_m = c_v T \tag{9.1}$$

The test problem, as Marshak described it, uses analytic opacity functions rather than empirical data. The scattering cross section is assumed to be zero and the absorption cross section goes as ν^s where s is a selectable parameter. For realistic treatments, setting $s = 3$ gives an analytic opacity that qualitatively resembles actual material opacities. This analytic opacity leads to a Rosseland mean opacity given by

$$\kappa_R = \frac{b}{T^s} \tag{9.2}$$

where b is a constant.

If we assume that the material energy density dominates the radiative energy density (i.e. $\mathcal{E}_m \gg \mathcal{E}$) and that there is no hydrodynamic motion (i.e. $D\rho/Dt = 0$), then the energy equation for equilibrium radiation diffusion[31],

$$\rho \frac{D}{Dt}(\frac{1}{\rho}(\mathcal{E}_m + \mathcal{E})) + \rho(\mathcal{P}_m + \mathcal{P})\frac{D}{Dt}(\frac{1}{\rho}) = \vec{\nabla} \cdot \frac{c}{3\kappa_R}\vec{\nabla}\mathcal{E} \tag{9.3}$$

can be re-written as

$$C_V \frac{\partial T}{\partial t} = \vec{\nabla} \cdot \frac{cT^s}{3b}\vec{\nabla} a T^4 \tag{9.4}$$

Pomraning[31] re-arranges this equation to the following form

$$\frac{\partial T}{\partial t} = K\vec{\nabla} \cdot T^n \vec{\nabla} T \tag{9.5}$$

where $n = s + 3$ and K is the constant

$$K = \frac{4ac}{3bC_V}. \tag{9.6}$$

The result is an analytic equation for material temperature that can be compared against simulation results. Unfortunately, Marshak's test problem was designed with an assumed analytic opacity that is very uncomplicated in energy space. It is this analytic opacity which makes the test problem so readily solvable with grey diffusion. Using multigroup diffusion with this analytic opacity will lead to negligible improvement.

Instead, to use the Marshak wave as a test problem for comparing static and adaptive multigroup diffusion, we must deviate from the description given by Marshak and use a realistic material opacity. By doing so, we lose the luxury of the analytic result to compare against, but the internally consistent figure of merit described in section 9.1 will still allow us to compare the two implementations of multigroup diffusion.

9.2.1 Multigroup Solution

What makes a multigroup radiation treatment different from a grey solution is the fact that photons in different energy groups may see dramatically different material opacities. If there is an energy group whose opacity is relatively low, the photons in that group will have a correspondingly high diffusion coefficient. Therefore, they will tend to propagate faster through the medium. Contrast this to the grey solution where all photons see the same opacity and thus are affected by the same diffusion coefficient.

The fact that some photon energies may stream away from their source relatively quickly introduces the possibility of non-equilibrium local photon distributions. This environment makes grey diffusion an inappropriate approximation and is a perfect example of where multigroup can be used to increase accuracy. A good test to demonstrate this difference is to look at the arrival time of photons at a particular point

some distance from the hot driving boundary in the Marshak wave simulation.

9.3 Static Multigroup and the Marshak Wave

100 eV Marshak wave

We set up a standard Marshak wave in an iron slab one centimeter thick. At $t = 0$ the slab is at an initial temperature 0.5 eV. The hot boundary is at a constant temperature of 100 eV. We monitor a point 0.5 centimeters into the slab and watch for the radiation energy density to rise with the arrival of the radiation wave. For a grey simulation there is only a single radiation energy density but in the multigroup simulation there will be a separate radiation energy density for each group. By monitoring this multigroup radiation energy density, we can examine the arrival time of energy versus the opacity.

First we do a grey diffusion run and look at the arrival time of the diffusion front at the 0.5 cm mark. With a starting material density of 0.005 g/cm^3, the grey radiation energy density reaches the midpoint of the slab at $5 \times 10^{-9}s$, see figure 9-1[1]. Note that the low density here was chosen to lower the overall material opacity such that diffusion speeds would be reasonably quick.

Now we compare this to the same data obtained when multigroup is enabled rather than simple diffusion. In this case, we have to make a decision as to how many energy groups to include. We start with a very simple case of ten energy groups. The eleven group boundaries are chosen to be uniformly distributed in log-energy space between the energies of 0.1 eV and 1 MeV. In this case, the grey radiation energy density arrival time is the same at $5 \times 10^{-9}s$, see figure 9-2.

Examining the plots of the grey radiation energy density with and without the multigroup engine enabled shows that multigroup preserves the behavior of the frequency-integrated radiation energy density. The two plots are identical to within the width of

[1]This plot (like many in this chapter) was generated using one of the standard output methods in RAGE, known as a Tektronics plot. The exponent in the upper-left is applied to the ordinate values. The thin bar across the top of the plot shows the location of the mesh boundaries. In this case, there are 100 cells uniformly distributed between 0.0 and 1.0 cm.

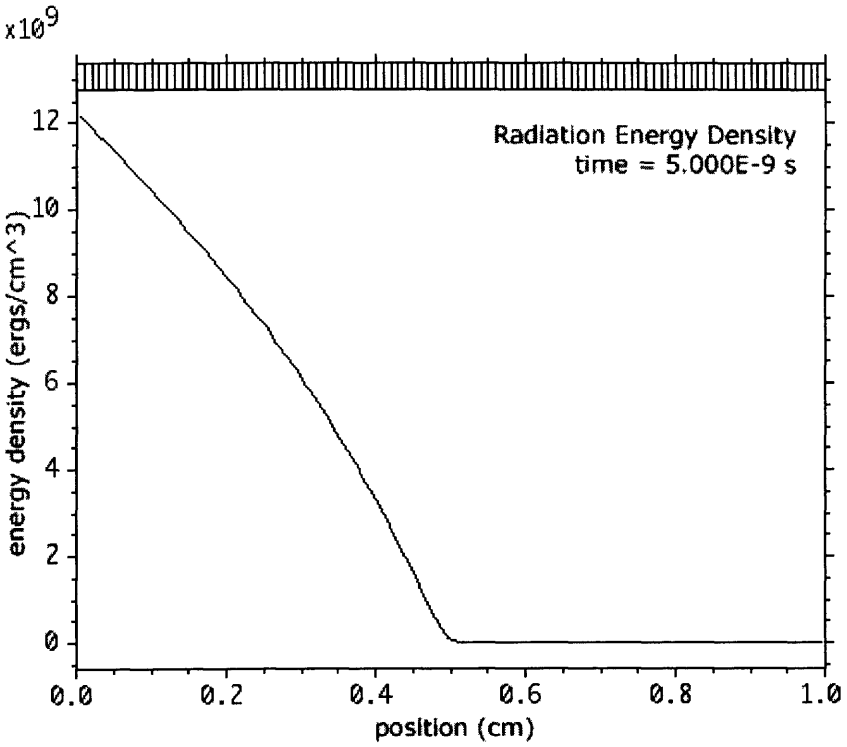

Figure 9-1: The radiation energy density as a function of position. In this example, the driving wall temperature is 100 eV.

the line. This is significant because it indicates that the grey diffusion approximation was sufficient to match the multigroup calculation's radiation energy density.

As long as the bulk of the radiation energy density is composed of photons with short mean free paths, the Rosseland mean opacity will preserve the integrated energy flux when used with the diffusion approximation. This means that, in regimes where the diffusion approximation is appropriate, using multigroup will not lead to a better calculation of the integrated radiation energy density. The fact that this is the case for the example above suggests that we are in an optically thick regime.

The boundary temperature of 100 eV is cold enough that the vast majority of photons have short mean free paths in the iron at a density of $0.005 g/cm^3$. Because of the Planck distribution of photons at a temperature of 100 eV, there are very few photons at energies high enough to see the iron slab as optically thin. This can be seen directly by examining the multigroup radiation energy density, i.e. the radiation energy density contained within each of the 10 energy groups defined in this example.

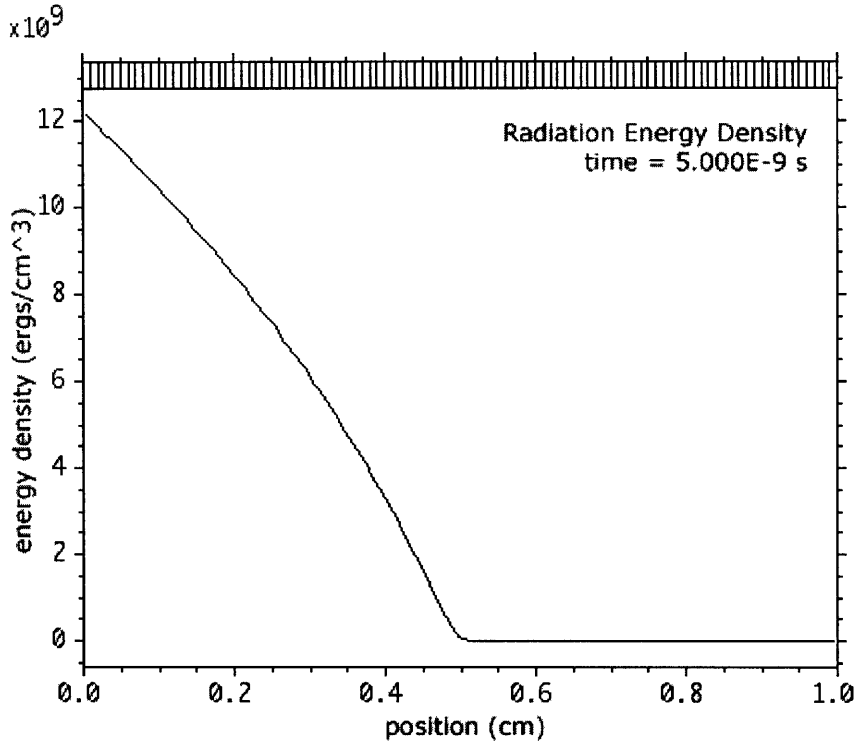

Figure 9-2: The integrated radiation energy density as a function of position for multigroup transport. In this example, the driving wall temperature is 100 eV.

If we look at one the highest energy groups, for instance the ninth group which spans the photon energies from 100.0 keV to 316.2 keV, we see data that is basically noise at the limit of machine precision. Effectively, there is no measurable radiation energy density in this group, see Figure 9-3.

This is understandable when we consider that the driving wall temperature in this Marshak wave is only 100 eV. We expect the photon energy density between energies represented by group 9 to be negligible[2]. Additionally, any photons in this energy regime would see a very low opacity and would likely stream out through the problem boundaries at the speed of light. For this example, the optical thickness is so low for

[2]To get a quantitative measurement of the radiation energy density between 100.0 keV and 316.2 keV for a system in equilibrium at T = 100 eV, we can integrate the Planck spectrum between the two photon energies. However, given the exponential fall-off of the Planck distribution and the fact that we are examining an energy band that is three orders of magnitude above the driving temperature, we expect a truly minuscule result. The factor $\exp(h\nu/kT)$ is $\approx 10^{434}$ for $h\nu = 1000kT$. This number, in the denominator, renders the integral undetectable to a computer with 8-byte floating point precision.

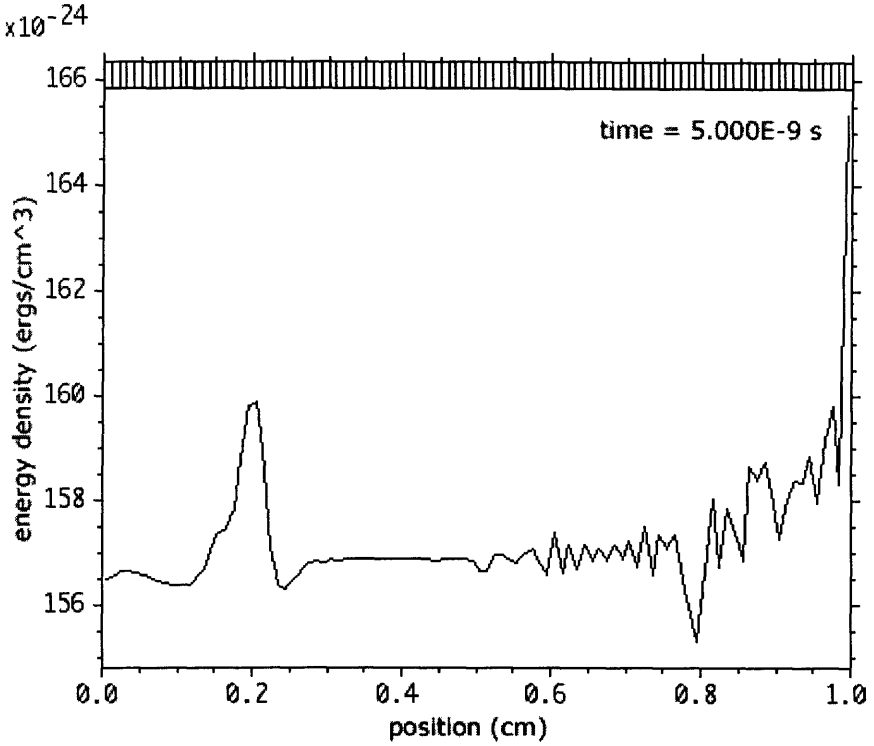

Figure 9-3: The multigroup radiation energy density in group 9, spanning photon energies from 100 keV to 316.2 keV. Note the exponent on the ordinate values—this is at the limit of machine precision and consists entirely of round-off artifacts.

high energy photons that groups 7-10 are all devoid of measurable radiation energy density.

At the other extreme of photon energy is the first energy group, with energies between 0.1 eV and 31.62 eV. Here the material opacity is enormous and the diffusion coefficient is correspondingly small. As a result, photons in this energy band diffuse incredibly slowly. As photons of higher energy diffuse through the iron slab and heat it, the thermal emissivity from the material will produce photons in this low-energy group long before they are able to diffuse out from the hot boundary. Therefore, we expect to see this energy group's multigroup radiation energy density in secular equilibrium with the material temperature.

Comparing the group's multigroup radiation energy density (figure 9-4) to the material temperature (figure 9-5) shows a qualitative similarity that bolsters this hypothesis. To get the quantitative proof that group one is indeed in secular equilibrium

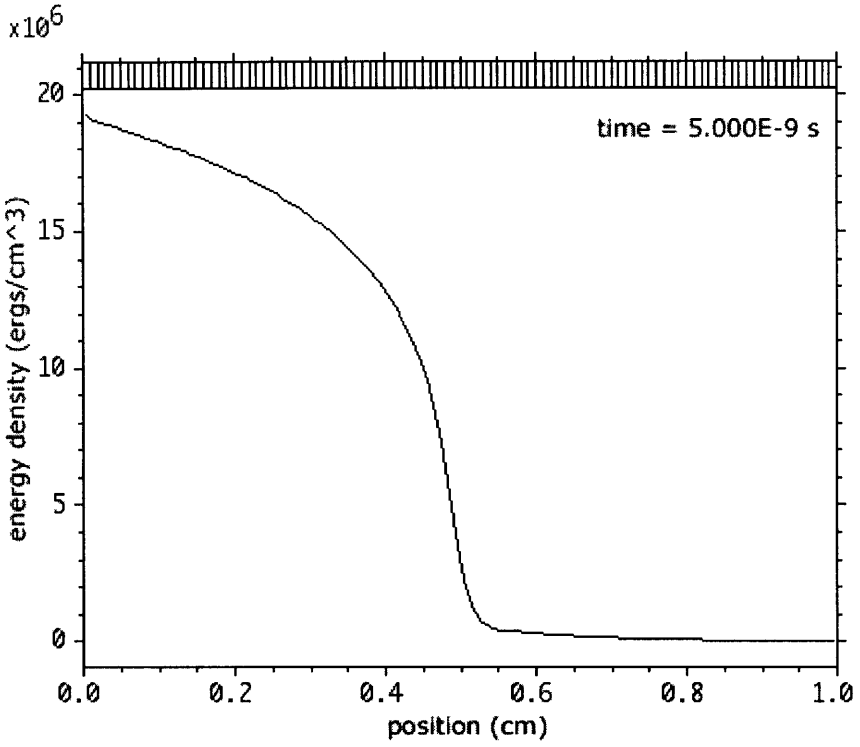

Figure 9-4: The multigroup radiation energy density in group 1, spanning photon energies from 0.1 eV to 31.62 eV.

with the material temperature, we can examine the "*color temperature*[3]" for group one. With the exception of a few cells right at the diffusion front, the group one color temperature is equal to the material temperature. This tells us that the group one radiation energy density is in equilibrium with the material and is being generated by emission from the material - not diffusion from neighboring cells.

Notice that the magnitude of the first group's multigroup radiation energy density is $10^7 ergs/cm^3$ whereas the total radiation energy density is three orders of magnitude greater (figure 9-2). Obviously these low-energy photons contribute only a small fraction of the total radiation energy density in this simulation.

If we examine higher and higher energy groups, the qualitative behavior transitions smoothly from the secular equilibrium of group 1 to free streaming in groups 7-10. The bulk of the radiation energy density is contained in groups 3 and 4 (42% and

[3]The color temperature of a group is defined as the temperature whose Planckian distribution would account for the radiation energy density in the group. In strict thermodynamic equilibrium, the color temperatures for all groups are equal to the material temperature.

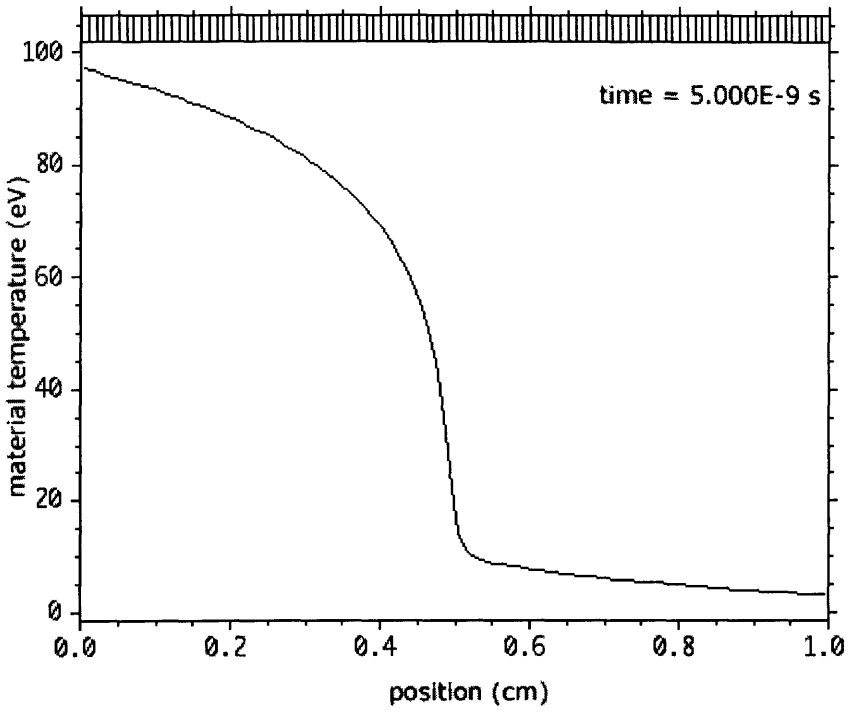

Figure 9-5: The material temperature at the time when the integrated radiation energy density propagation front reaches 0.5 cm.

54% of the total, respectively). The remaining 4% is almost all in low-energy group 2. Aside from trivial exceptions, the radiation energy density in this simulation exists in photons with energies between 31.62 eV and 1 keV.

As was shown in the comparison of figures 9-1 and 9-2, the frequency-integrated radiation energy density is all being described well by grey diffusion. Therefore, the material opacity between 31.62 eV and 1 keV at this density is high enough to warrant a diffusion description. Unfortunately, when this is the case, the placement of groups in a multigroup radiation treatment has no bearing on the accuracy. In order to demonstrate the effects of the adaptive multigroup radiation diffusion algorithm, we must change the environment of this simulation to push the radiation energy density into lower-opacity regimes. The two ways to do this are to lower the material density, or to increase the driving wall temperature.

500 eV Marshak wave

In an attempt to drive the radiation energy density into regions of lower material opacity, we create a new Marshak wave problem with a driving wall temperature of 500 eV. Proceeding as before, we do a grey diffusion simulation and look for the point in time when the radiation front reaches the midpoint of the slab (the solid line in figure 9-6).

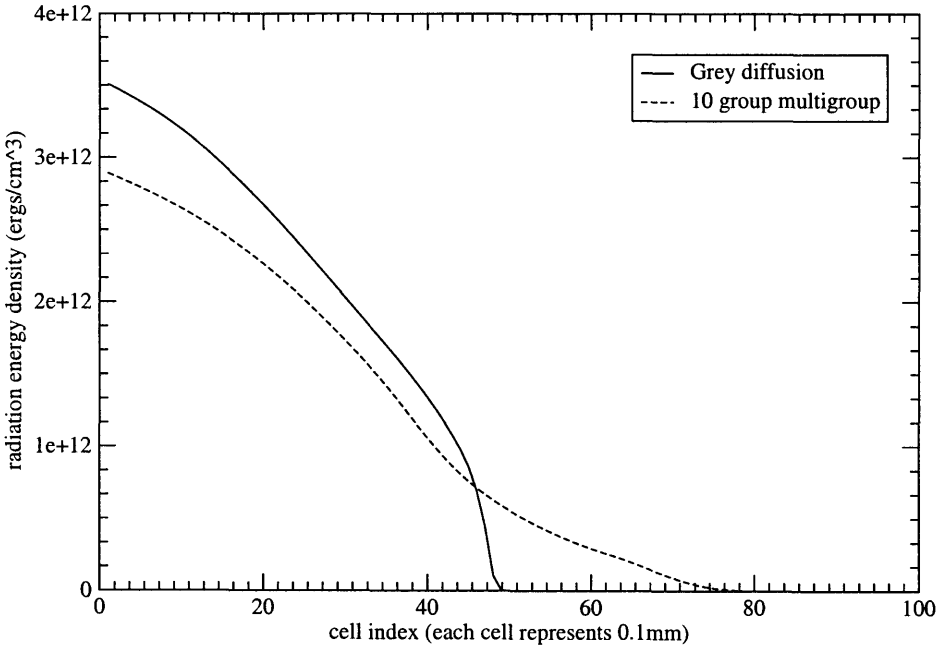

Figure 9-6: The frequency-integrated radiation energy density for a Marshak wave with a 500 eV driving wall temperature, showing the difference in output for grey and 10-group multigroup diffusion simulations. The time required for the grey solution to propagate half-way through the slab, as shown in this plot, was 2.4×10^{-11} s.

Comparing this result to figure 9-1 reveals some interesting physical differences between the 100 eV and 500 eV grey Marshak waves. The first is that the diffusion speed is considerably higher with the increased wall temperature. The apparent diffusion speed, based on the time it took to propagate 0.5 cm, for the 100 eV wall temperature was 0.33% c. For the 500 eV wall temperature it is 68% c. Clearly, the photon mean free paths are considerably longer with the new photon distribution.

Also apparent is the increased magnitude of the radiation energy density. The basic qualitative behavior is similar, but the value has increased by two orders of magnitude. This result is expected, considering the functional form of the Planck distribution (equation 3.13). Increasing kT significantly makes the factor of $\exp(h\nu/kT)$ much smaller. This term appears in the denominator of the Planck function, and therefore results in an increase in photon density (and therefore radiation energy density) at all photon energies.

For the 100 eV driving temperature example, the radiation energy density didn't change significantly by enabling multigroup. When we turn the multigroup engine on for the 500 eV case, we get the result given in the dashed line in figure 9-6. The evolution of the integrated radiation energy density is significantly different with multigroup enabled. The value at the wall has come down by 17% and there is a large precursor diffusing faster than the rest of the photon energy. The propagating diffusion "front" seen in the 100 eV case has become less distinct. This is caused by the different opacities seen by each energy group. The result is different diffusion coefficients for each group and therefore different speeds of propagation. When the frequency-integrated radiation energy density is calculated, these differences in propagation speed cause the dispersion of the diffusion front, as seen in the dashed line in figure 9-6.

As with the 100 eV example, we can calculate the percentage of the total radiation energy density present in each group. In this way, we can determine if any of the energy groups are being wasted on regions of energy space that do not contribute to the evolution of this problem. The MGMFG package in RAGE spends an equal amount of time on each energy group, regardless of whether the photon count in that group is negligible. We would like to cull any groups that have no radiation content. Alternatively, we would like to pull in the energy limits of the problem to avoid very low and very high photon energies that are not having an effect in the simulation. The following table shows the distribution of radiation energy density by group:

Group	Percentage
1	<0.01%
2	0.4%
3	1.1%
4	19.4%
5	54.2%
6	24.9%
7	<0.01%
8	≪0.01%
9	≪0.01%
10	≪0.01%

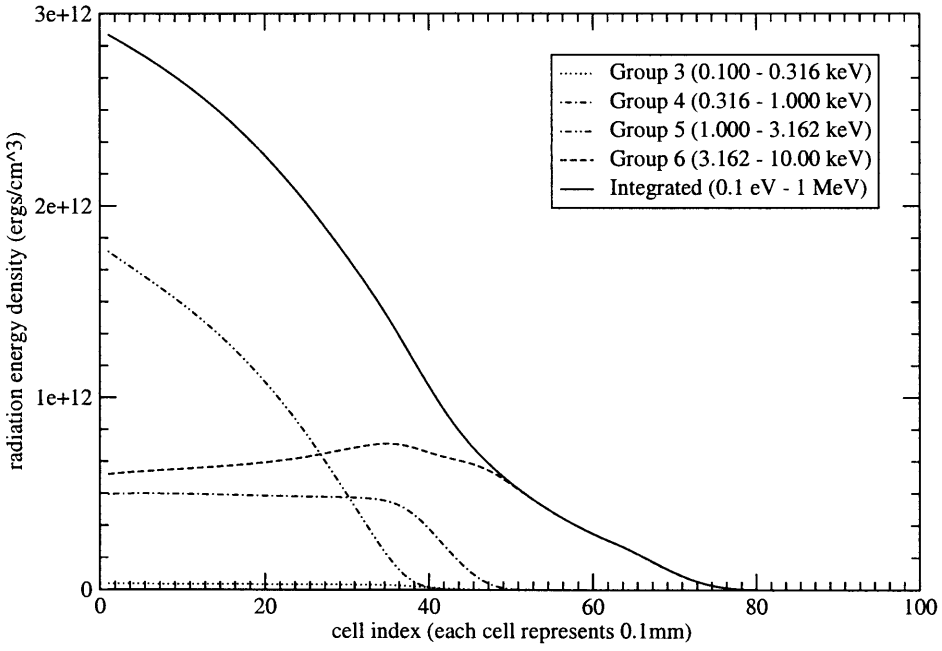

Figure 9-7: The multigroup radiation energy densities for a 10-group Marshak wave with a 500 eV driving wall temperature, showing each group's contribution to the integrated total. The numbers in parenthesis next to each group label are the group boundaries for that group. All ten groups are plotted on the same scale; those not listed in the legend contributed too little to appear.

By the time the diffusion front has progressed half way through the slab, nothing

has happened which has caused any measurable radiation to appear in groups 8, 9, and 10. These groups are therefore not affecting the evolution of this simulation. Furthermore, groups 1 and 7 contribute very little to the behavior of this Marshak wave. The contributions of each group to the integrated total are shown graphically in figure 9-7. Notice that the integrated radiation energy density is almost entirely made up of groups 4, 5, and 6. Groups 1, 2, 7, 8, 9, and 10 contribute so little that they cannot be seen on the scale of this plot. Also, notice that the entire precursor foot exists only in group 6.

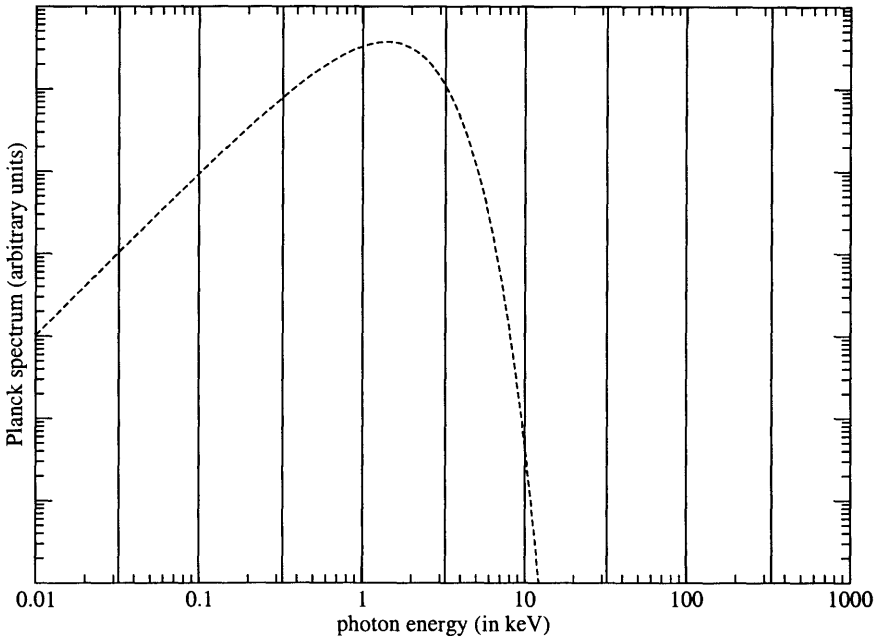

Figure 9-8: The Planck spectrum for a 500 eV wall temperature. The vertical lines are the group boundaries used in the initial 500 eV Marshak wave simulation.

The reason that so many groups are effectively not used is because the energy limits were chosen arbitrarily and do not reflect the likely radiation spectrum of this problem. Figure 9-8 shows the Planck emission spectrum for the 500 eV wall temperature overlaid with the group structure that was chosen. Clearly the Planckian contribution in the last three groups is expected to be insignificant. Group six is the highest energy group with any meaningful intensity in it, and thus we expect it to

have the lowest mean opacity (assuming a general ν^{-3} dependence). This is why group six encompasses the entire precursor foot in this Marshak wave.

It would be prudent to modify the group boundaries for this simulation such that our 10 groups more effectively span the active range of photon energies. Based on the above data, a good place for the lower energy limit would be somewhere below the current second group (31.62 eV). However, moving the lower energy limit above the initial cold material temperature will cause an error in RAGE. We can either put the lower energy limit at 0.5 eV or raise the cold slab temperature somewhat. The upper energy limit should be placed above the current sixth group (10 keV).

We can now move the energy limits and recalculate the uniformly-spaced group boundaries and associated multigroup opacity table. The above table and plot suggest that we will not be reducing accuracy, as we will only be neglecting photons that have no measurable effect on the outcome of the simulation. However, this is not necessarily the case. Even though the new energy limits still span all of the relevant photon energies, the location of the boundaries themselves relative to the material opacity have changed. A strong gradient feature in the opacity might move from being on a group boundary to being in the middle of a group, thus drastically changing the relevant group mean opacities. As an example, consider figure 9-9, which shows the results of changing the upper energy limit.

It is important to point out that the difference in energy span between these two simulations is entirely composed of energy space that was shown above to contain negligible radiation energy density. The significant differences in the two data sets are simply the result of changes in the position of group boundaries relative to features in the material opacity.

The question arises of how to determine which of these two radiation energy density data sets is more accurate. As was described in section 9.1, a good method for making this determination is to compare both data sets to a high-resolution data set made with significantly more energy groups. The assumption is that by having very thin energy groups, the output becomes less sensitive to the precise placement of the energy boundaries. To test this assumption, we run the 500 eV Marshak wave

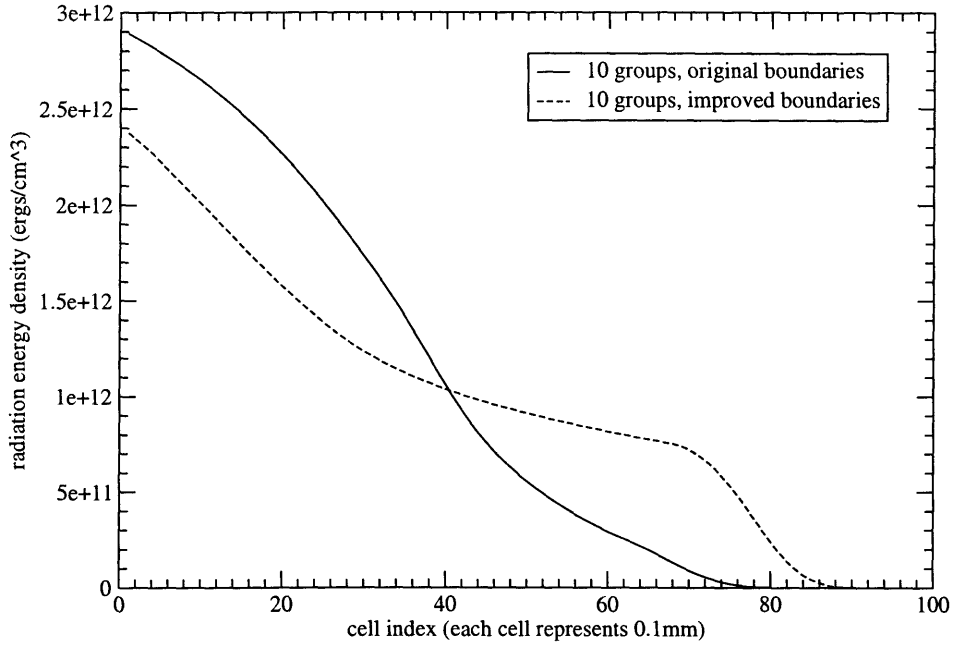

Figure 9-9: The frequency-integrated radiation energy density for two ten-group simulations with different high-energy limits. The solid line data has a high energy limit of 1000 keV. The dashed line has a high energy limit of 30 keV. Both high energy limits are in a region of energy space which contains negligible photon populations.

simulation with 1000 energy groups uniformly distributed between the original energy limits of 0.1 eV and 1 MeV. The radiation energy density data is then compared to a similar run with energy limits of 0.5 eV and 70 keV. The two data sets are equal to within the line width in figure 9-10. This demonstrates that the output is considerably less sensitive to changing the group boundary positions when the group widths are small.

Figure 9-10 shows the high-resolution data set overlaid on the data from figure 9-9. It is clear that one of the 10-group data sets matches the high resolution data much better and is thus more accurate. We can get a quantitative measurement of the similarity by calculating the RMS differences between each of the two low resolution data sets and the high resolution data set. What is surprising about this plot is the fact that the "improved" energy limits led to the worse answer. Clearly, with

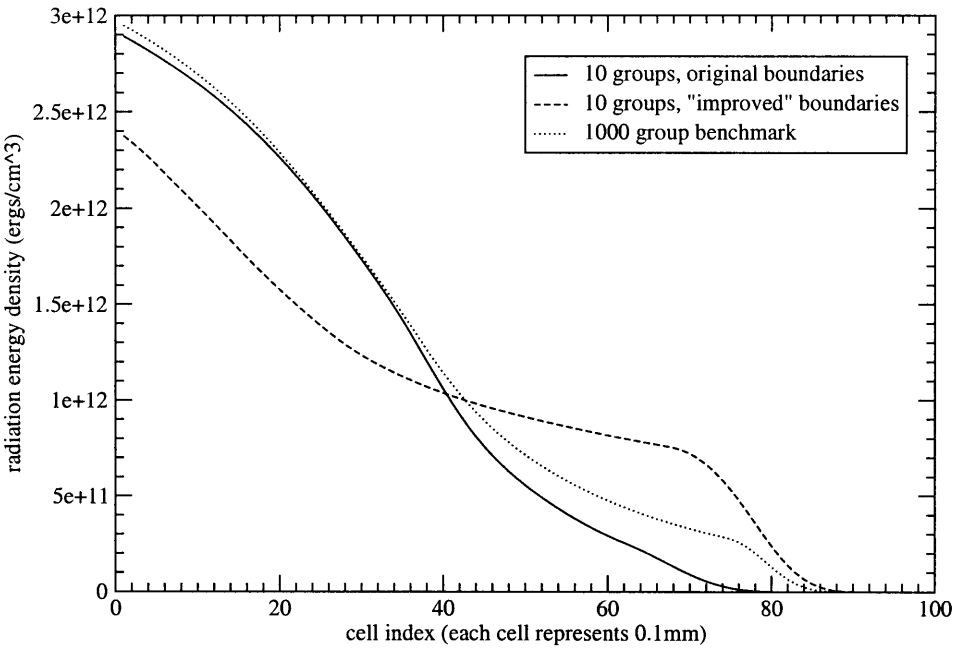

Figure 9-10: The frequency-integrated radiation energy density for two ten-group simulations with different high-energy limits is compared to a high-resolution 1000-group simulation. The solid line data is the 10-group run with an energy limit of 70 keV. The dashed line is the 10-group run with an energy limit of 30 keV. The dotted line is the 1000-group high resolution data.

only ten groups, having slightly thinner energy groups is not sufficient to overcome the highly non-linear interaction between the group boundaries and features in the material opacity.

By running the 500 eV Marshak wave simulation repeatedly and varying the upper energy limit, we can measure the wide variance in accuracy that can come about simply by moving the group boundaries. Note that these RMS differences are normalized to the RMS value of the high-resolution data.

High energy limit	RMS difference
30 keV	0.077
50 keV	0.178
70 keV	0.282
100 keV	0.216
500 keV	0.236
1000 keV	0.836

Two things are notable about this table of information. First is that the RMS difference does not vary monotonically with the energy limit. This is expected, as we are varying the location of the energy group boundaries on a highly non-linear material opacity. More surprising is the fact that the original high energy limit of 1 MeV has one of the lowest RMS differences, and we know from previous data that this simulation wastes three of its ten groups.

This is more than just a curiosity — it is exceptionally significant. If the last three groups in this calculation are truly not contributing, then removing them altogether should return the same RMS difference. We can calculate the uniformly-spaced group boundaries from 0.01 eV to 1 MeV, then simply remove the last three boundaries and run a 7-group test. The upper energy limit becomes 31.62 keV. In this case the RMS difference comes out the same to six significant figures.

What we have accomplished here is to choose our group boundaries such that we get a better answer with seven groups instead of ten. In theory this represents a 30% reduction in computation time for the same answer. In actuality there are components to RAGE that take time to execute beyond the MGMFG engine. This overhead reduces the actual computational savings in this case to 21%, but this is still a significant gain.

Clearly, there is much to be gained by carefully considering the location of the energy group boundaries. However, the method used above of moving the upper boundary around in wasted energy space looking for a low RMS is clumsy. Instead, we now examine the opacity curve itself and attempt to hand-pick boundary locations.

9.3.1 Manually choosing photon energy group boundaries

The accuracy afforded by a given set of group boundaries is based on the relative position of the boundaries and features in the monochromatic opacity. It is tempting to examine the monochromatic opacity and place the group boundaries manually at significant features such as the K-edge and regions of dense bound-bound transition lines. However, as discussed in section 6.3, the opacity can change dramatically due to heating or changes in density. For the 500eV Marshak wave, we see a dramatic increase in temperature. Figure 9-11 shows the opacities for the initial cold slab and the equilibrium temperature of the driving wall.

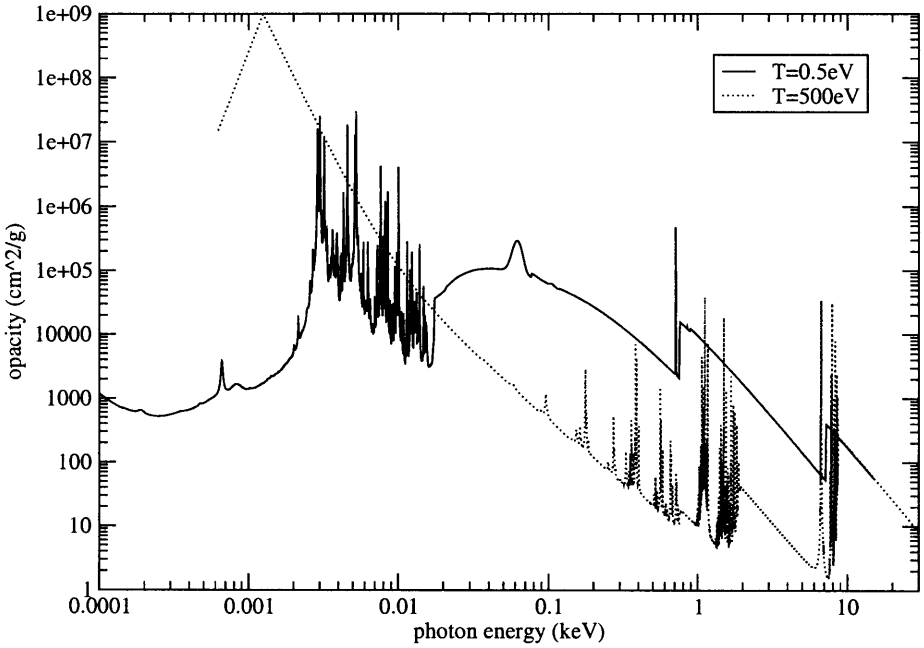

Figure 9-11: The opacity for iron at a density of 0.005 g/cm^3 and two different temperatures. These temperatures represent the starting "cold" slab and the equilibrium temperature of the hot driving wall. The two opacity data sets have different limits as is shown here. The TOPS code extrapolates along a curve of ν^{-3} for opacities off the table.

The T = 500 eV opacity follows ν^{-3} reasonably well between photon energies of 1 eV and 100 eV and would thus require relatively low group resolution in this

regime. However, this is exactly the regime that contains the bulk of the bound-bound transition lines in the T = 0.5 eV opacity. Also the apparent location of the L-edge has shifted 1.3 keV during the change in temperatures. Clearly, these two opacities have different optimal group placements.

One could place boundaries at all optimal points for both curves, but this ignores that fact that the material will be be continuously heated from one temperature to the other and the actual opacity will evolve as well. For a steady-state problem, manual choice of energy group boundaries is feasible. For a simulation whose temperature, density, or material composition evolves with time, the manual selection would have to be undertaken repeatedly. The adaptive multigroup radiation diffusion method provides a means to automate this process.

9.3.2 Sensitivity of results to group boundary placement

In section 9.3, we demonstrated that the results of a multigroup radiation diffusion simulation were very sensitive to the location of the energy boundaries. This sensitivity arose as a result of changing the energy limits while keeping the number of group boundaries constant and keeping them uniformly spaced. The result was group boundaries which shifted slightly in energy space from run to run and widely varying results.

We expect this same shifting of group boundaries to take place if we keep the energy limits constant but change the number of groups (and still maintain uniform group spacing). This is troubling because it means that we might get unpredictable changes in accuracy by adding groups, despite the conjecture that adding groups to a multigroup simulation increases the accuracy of the results. To test this hypothesis, we set up another 500 eV Marshak wave simulation and run it many times, increasing the number of groups on each run. The outcome of each run is compared to the 1000-group benchmark and plotted in figure 9-12.

There is an obvious trend of improvement as the number of groups is increased. However, the plot is anything but monotonic. At half of the points on this plot, adding a single group leads to a worse answer. This unpredictability is a result of the

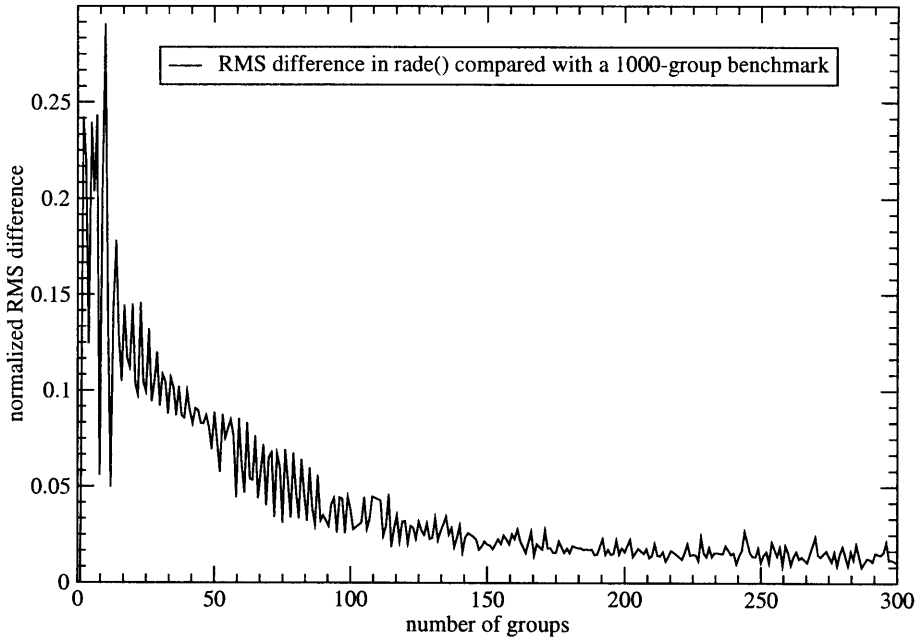

Figure 9-12: The normalized RMS error in radiation energy density as a function of group count.

group boundaries moving with relation to the features in the material opacity.

The magnitude of the oscillations drops off as you increase group counts and out towards 1000 groups it represents negligible changes. However, as stated previously, we tend to run large simulations with very small group counts due to constraints on computational availability. As can be seen in figure 9-12, simulations with the smallest group counts have the largest swings in accuracy.

To further illustrate the extent of this unpredictability, we change the energy limits slightly and run the same series of simulations again. Figure 9-13 shows the same plot for two different sets of energy boundaries and zoomed in on just the small group count regime. This graph tells us a great deal about the reliability of static multigroup radiation diffusion. For starters, both data sets exhibit wild oscillations in accuracy as the number of groups is increased. Furthermore, a group count that lends a very good accuracy for one set of energy boundaries frequently lends a very poor result for the other. At the most extreme separation, the RMS error in one plot

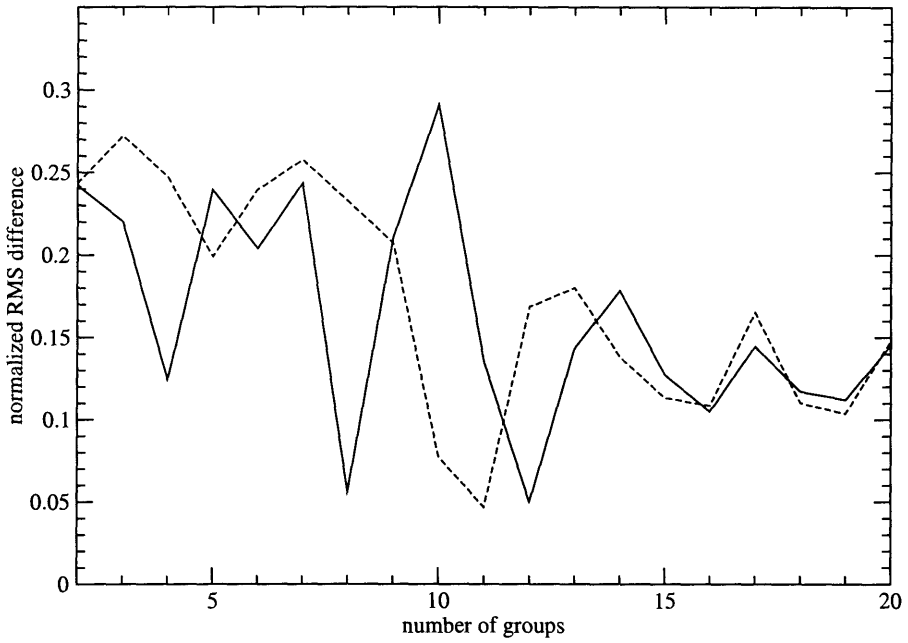

Figure 9-13: The normalized RMS error in radiation energy density as a function of group count. Comparison of two different energy limits at small group counts.

is 400% greater than the other.

When we are dealing with such a coarse group structure, we have no guarantee that adding an additional group will give us a better answer, and there is a reasonable probability that the answer could be significantly worse. The group counts that lead to particularly high or low accuracies are not constant over changing energy limits and thus cannot be predicted *a priori.*

The left-most point in figure 9-13 represents a single group or grey diffusion. The two data sets match at this point because the single group covers the entire active energy range. What is disturbing is that both plots contain points that are *less accurate* than the grey result. The solid line has a particularly low accuracy point all the way out at ten groups. What this indicates is that if you choose an unlucky set of energy limits you are not guaranteed of getting any improvement at all over grey diffusion with as many as ten groups and possibly more. Bear in mind that the ten group simulation takes approximately an order of magnitude longer to run than grey

diffusion.

The sensitivity to group boundary placement in static multigroup radiation diffusion is significant. At small group counts, there is no guarantee of improvement and a significant risk of a decrease in validity. There are certainly combinations of group count and energy limits that lead to a significantly improved answer, but there is no metric for choosing such a combination *a priori*. In this case we have the luxury of a 1000-group benchmark data set to test the accuracy of our various simulations. Unfortunately, the whole reason for running small group count simulations for large problems is because we cannot afford to run more groups. Thus, when it matters most, we cannot demonstrate that small group count static multigroup diffusion has produced an answer worth its increase in computational expense.

The fact that small group count static multigroup diffusion is so unreliable has led to it being set aside, and grey diffusion used in its place. This is true even in simulations where the radiation field is known to be out of equilibrium with the matter — and therefore that grey diffusion is physically inadequate for solving the problem. The added computational expense of multigroup is simply deemed not worth the risk of a possibly worse answer.

9.4 Adaptive Multigroup and the Marshak Wave

It is the unreliability and effective uselessness of small group count static multigroup that was the motivation for adaption in energy space. If the source of the sensitivity is the relative placement of group boundaries and features in the opacity, then tracking those features and moving the group boundaries to match should alleviate the problem.

By enabling the adaptive multigroup engine, we are presented with a simulation which automatically repositions the group boundaries at each time step based on the current opacities present in the problem. Having group boundaries at varying positions makes it difficult to do a comparison of any group-specific parameter. For this reason, we will continue to use the frequency integrated radiation energy density when

comparing adaptive vs. static multigroup tests as well as for various incarnations of the adaptive multigroup method against each other.

9.4.1 Parameter study for adaptive multigroup

The implementation of adaptive multigroup radiation diffusion in RAGE has two main parameters. The most significant one is the number of groups in the high resolution IPCRESS table. This table is indistinguishable from the IPCRESS table that is used in static multigroup simulations, except that it is assumed to contain more energy groups than will actually be used. The IPCRESS table contains the total and scattering opacities for all materials in the problem on a user-supplied grid of temperature and density points. For all of the following data, we will hold the temperature and density points (as well as the materials) constant. The only remaining aspects of the IPCRESS data that might vary from simulation to simulation are the number of energy groups and the location of their boundaries. As was demonstrated in section 9.2.1, for suitably thin energy groups, the boundary placement is irrelevant. For the data in this section, we will fix the upper and lower energy limits to 50 keV and 0.5 eV, respectively.

The only other user-selectable parameter in the adaptive multigroup engine is the number of groups to coarsen down to. This is the number of groups which is handed off to the MGMFG radiation diffusion engine, and the number which determines the computational expense.

In the following sections, we will vary both the number of high-resolution groups as well as the low-resolution group count and show how the results change, and how the results compare to their static-multigroup equivalents.

9.4.2 Varying the number of coarse groups

In this section, we keep the IPCRESS table constant — all simulations have 1000 high-resolution energy groups. We vary the number of coarse groups and see how the integrated radiation energy density is affected.

As an initial test, we set the number of coarse groups to 1000. Because the number of coarse groups is equal to the number of fine groups, the adaptive multigroup engine will return the same group structure every time and the results should be the same as a static 1000 group simulation. Examining the RMS difference between the integrated radiation energy density at the same point in time for static multigroup and "no-reduction" adaptive multigroup shows that they are equivalent. As with previous sections, this 1000-group simulation will be used as the benchmark against which all other tests will be compared.

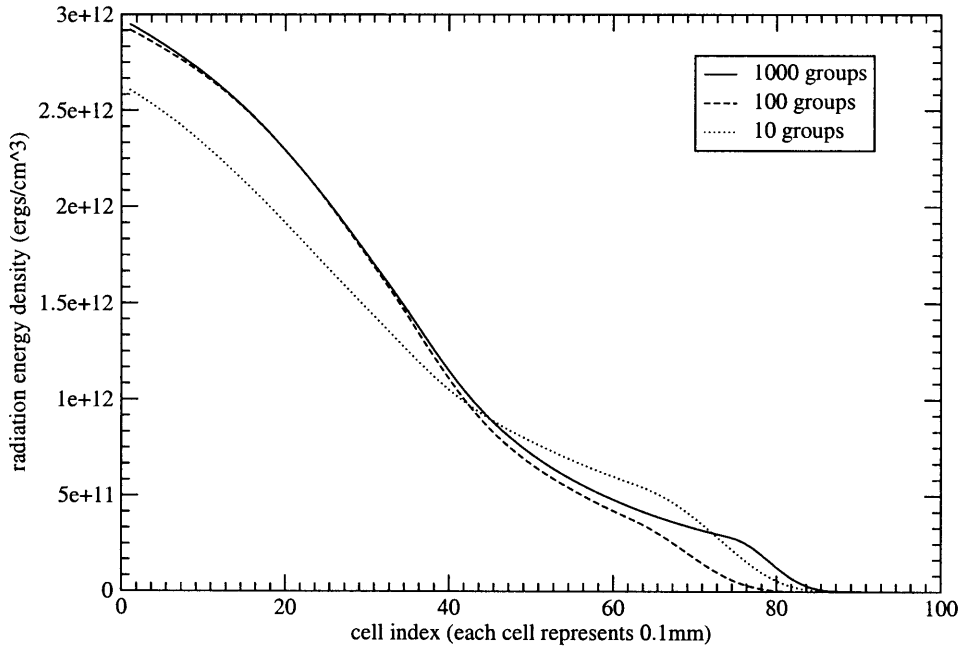

Figure 9-14: An examination of the falloff in accuracy as the number of coarse groups is decreased. The number of high-resolution groups and the energy limits are held constant.

Figure 9-14 shows the effect on the integrated radiation energy density as the number of coarse groups is decreased. As expected, the accuracy decreases as the number of coarse groups is decreased. The quantitative results are tabulated below, again normalized to the RMS value of the benchmark result:

# Coarse groups	RMS difference
1000	0.000
500	0.002
250	0.005
100	0.013
50	0.018
25	0.034
10	0.044

The data in the table show that as you increase the number of coarse groups, there is a diminishing return on accuracy gained.

9.4.3 Varying the number of high-resolution groups

In this section we keep the number of coarse groups the same and vary the number of high-resolution energy groups. For this investigation we choose an arbitrary number of course groups, 50. The number of fine groups is varied from 1000 down to 50. Again, the results are compared against a 1000-group static multigroup simulation.

The table below shows the effect of reducing the available group boundaries to choose from when coarsening. The reduction in accuracy is intuitively obvious, as the group coarsening routine is no longer necessarily able to select the optimal group boundary location. Also, as the high-resolution energy groups become less numerous, they also become wider. As was shown previously, this makes the accuracy more prone to the exact position of the various high-resolution boundaries.

Because the number of coarse groups was held constant throughout this series of tests, the computation time for each test was constant. The number of coarse groups determines the amount of work that must be done by the MGMFG engine. The number of high-resolution groups determines the size of the master IPCRESS table in memory. The number of high-resolution groups that can be included is only limited by the available memory of the computer.

# High-resolution groups	RMS difference
1000	0.0177
500	0.0351
250	0.0676
100	0.0802
50	0.0886

As can be seen in the tabular data above, which was normalized to the RMS value of the 1000-group results, the benefit of adaptive multigroup is negligible until the number of high resolution groups gets to be significantly higher than the number of coarse groups. In the last row, we are dealing with a high-resolution table that has the same number of group boundaries as the low-resolution table. Therefore, in the 50 group case, the adaption algorithm has no choice on where to place the boundaries and effectively does nothing. The RMS does not begin to rise significantly until we approach 500 groups — an order of magnitude larger than the coarse group structure.

Of course, simply adding memory to the computer and going to an obscenely high number of high-resolution groups does not provide for limitless accuracy. This is a simple consequence of the complexity of the representative opacity. For realistic problems, a fixed number of groups will always leave some amount of opacity structure in each group and will thus lose information during the creation of the group mean opacity.

Also, as was shown in section 9.3, choosing slightly different group boundaries in the 1000-group simulation led to a negligible change in accuracy. Therefore, above a certain number of high resolution groups, we expect the benefits of adding more high resolution groups to drop off.

As with static multigroup, the number of coarse groups is the primary user-selectable parameter in determining the accuracy of the simulation. However, in order to maximize the gains in accuracy by enabling adaption multigroup, the high-resolution group count should be significantly higher than the coarse group count.

9.4.4 Adaptive vs. Static Results

We are primarily interested in the performance of adaptive multigroup radiation diffusion at low coarse group counts, where static multigroup diffusion is unreliable.

In this section, we compare the results of small group count static and adaptive multigroup test problems against a 1000-group static benchmark. To be generous to the static multigroup, we'll start with the best case energy limits as given in the table in section 9.3: 0.5 eV lower limit and 30 keV upper limit.

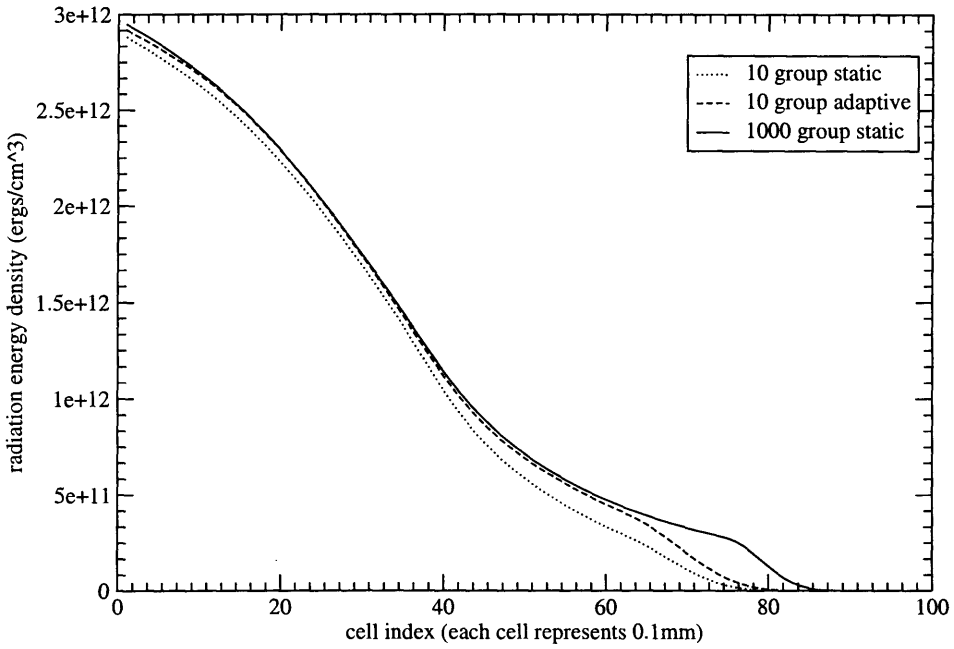

Figure 9-15: An examination of the improvement in accuracy available by enabling adaptive multigroup for the 500 eV Marshak wave. These simulations were conducted at the best-case energy limits for the 10-group static multigroup data.

Figure 9-15 shows that, even in the best-case scenario for static multigroup, the adaptive multigroup algorithm offers some improvement. The normalized RMS difference for the 10-group static simulation compared to the 1000-group static simulation is 0.0774. The normalized RMS value for the 10-group adaptive simulation compared to the 1000-group static simulation is 0.0443 - a reduction of 43%.

Of more practical interest is to measure the added accuracy in terms of computa-

tional time saved. In order to match this improved accuracy with static multigroup and the same energy limits, we would have to add four additional groups (giving a normalized RMS difference of 0.0488). The number of groups factors linearly into the computational expense of the MGMFG package. The 14 group static simulation and 10 group adaptive simulation achieve similar accuracy, and thus we can say that in this case the adaptive multigroup method saves 29% on computational time (assuming no overhead).

As a comparison, we now turn to the worst-case scenario from the table in section 9.3, the 70 keV upper energy limit. In this case, we found earlier a normalized RMS difference for the 10-group static simulation of 0.2821. The added energy space in this simulation (between 30 keV and 70 keV) goes largely unused by the radiation energy density, as we demonstrated earlier. As a result we expect to find the adaptive multigroup engine not wasting any group boundaries in this region. It is likely that the adaptive boundaries will be roughly the same as the previous case except that the last group will expand to encompass the additional dead space.

Figure 9-16 shows that in the worst-case scenario for static multigroup, enabling adaption in energy space returns an enormous gain in accuracy. The normalized RMS difference between the 10-group adaptive run and the 1000-group static run is 0.0451. As predicted, this is very similar to the adaptive simulation with less dead space inside the upper energy limit. The static simulation was significantly off the mark, and the adaptive RMS difference represents an 84% reduction. To match this accuracy with a static simulation, we are forced to increase the group count to 26. The reduction in computational cost in going from a 26-group static simulation to a 10-group adaptive simulation is an impressive 62% (ignoring overhead).

The preceding two examples show that by varying the energy limits, the static results can very widely while the adaptive results remain effectively constant. This is only the case as long as the energy limits remain outside of the active energy space. However, as was shown in figure 9-8, determining what photon energies are likely to be contributors to a given simulation is simply a matter of knowing the likely material temperatures. This is generally known *a priori* and thus setting the energy limits in

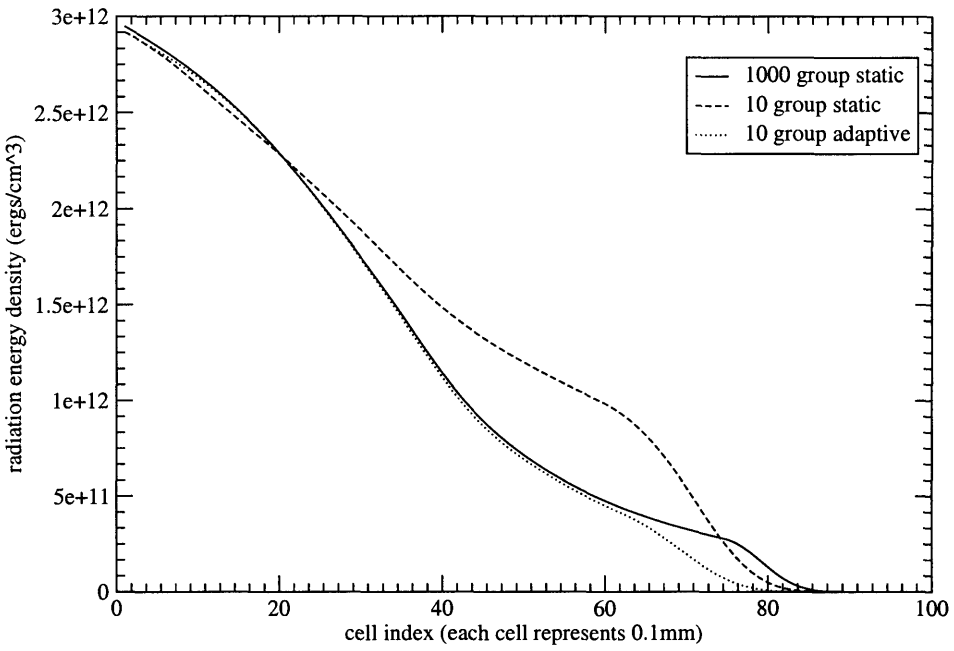

Figure 9-16: An examination of the improvement in accuracy available by enabling adaptive multigroup for the 500 eV Marshak wave. These simulations were conducted at the worst-case energy limits for the 10-group static multigroup data.

a reasonable location is straightforward. As was demonstrated above, exactly where they are placed is inconsequential when adaptive multigroup diffusion is used.

Effectively we have shown that the sensitivity to energy limit location found in static multigroup is eliminated with adaption enabled. The other sensitivity found in static multigroup is the sensitivity to group count. This is the more troubling of the two, as it was shown in section 9.3.2 that sometimes adding groups can lead to a significantly worse answer — even compared with the grey diffusion result.

To examine the effects of group count on the accuracy for small group count adaptive multigroup radiation diffusion simulations, we replicate figure 9-13 but with a third data set showing the adaptive results. The adaptive multigroup diffusion data in figure 9-17 shows a monotonic increase in accuracy as group count is increased from 1 to 20. The accuracy of the adaptive plot stays consistently above that of either static plot. The best-case data points for static multigroup come very close to the adaptive

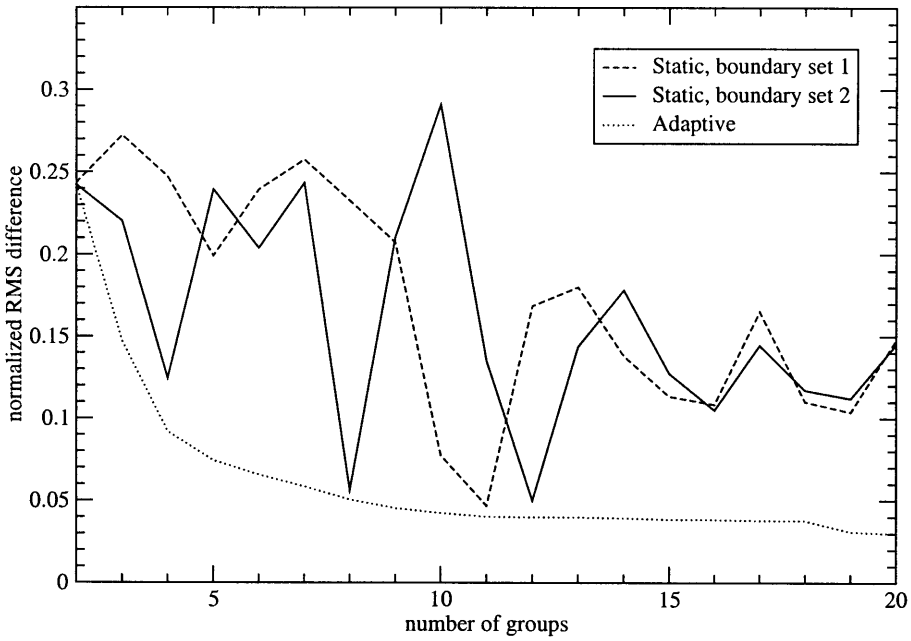

Figure 9-17: Two static multigroup diffusion data sets compared against an adaptive multigroup diffusion data set. The best accuracy points on both static plots still have higher error than the corresponding adaptive multigroup data points.

result, but in general the separation between the two is significant.

This result has profound implications for the use of multigroup radiation diffusion on large simulations where only small group counts can be afforded. Enabling adaption in energy space leads to an answer that is consistently more accurate than static multigroup diffusion. Furthermore, there is a predictable and reliable increase in accuracy when increasing the number of groups even in this very small group count regime. Because this is true, the users of RAGE can choose the number of coarse groups to run based on a realistic computer time budget and expect to get more accuracy as computational expenditures increase.

Chapter 10

Conclusion

10.1 Future Investigations

10.1.1 Performance Upgrades

The implementation of adaptive multigroup radiation diffusion within the RAGE code framework is a proof-of-concept code. No effort was made to optimize for computational speed or economy of memory usage. While the results shown here are very promising, the speed and memory overhead are still significant.

The bulk of the memory overhead involved with adaptive multigroup is filled by the master radiation energy density table. Currently the radiation energy density refinement that occurs after the radiation solver has updated the coarse radiation energy density uses a piecewise linear interpolator to map changes back into the master table. It is possible that this interpolation could be made more complex and the information then kept at the coarse level, requiring slightly more computational time but significantly less memory.

The radiation energy density table refinement is also the costliest routine in terms of time. Currently this routine is not coded in a manner that takes advantage of the massively parallel nature of the supercomputers on which it runs. By having each processor calculate changes to the multigroup radiation energy density independently, the cross-processor communication time could be reduced to a single global collection

of deltas followed by a single global dispersal of the cumulative changes.

10.1.2 Full RAGE integration

In addition to these performance shortcomings, there were compromises made in the functionality of RAGE in order to get the adaptive multigroup routine to a proof-of-concept state as quickly as possible. Major functional blocks of the RAGE framework were trimmed from the code so as not to unnecessarily complicate the initial design of the algorithm. The result is that simulations that require the use of these excluded modules will not work if adaptive multigroup is enabled. The list of features that currently break the adaptive multigroup module include spatial adaptive mesh refinement, hydrodynamics, multiple materials, user-defined (non-Plankian) radiation sources, and analytic material opacities. None of these features in RAGE is fundamentally incompatible with adaptive multigroup radiation diffusion. There is simply a certain amount of code that must be written to handle the effects of these features on the mechanics of energy-space adaption.

The AMR system in RAGE continuously adds and subtracts cells from the spatial mesh, and the multigroup radiation energy density (and its high-resolution master table) are defined on this mesh. Therefore, routines must be added to the adaptive multigroup module which allow it to refine and coarsen the multigroup radiation energy density variables correctly when AMR deems it necessary.

Currently the creation of the representative opacity assumes a single material. Allowing for multiple materials will involve simply adding a loop over all materials in each cell within the adaption routine. The weighting function for each material's opacity will have to be properly scaled by the mass or volume fraction of the material in the cell. The result will be a more complicated representative opacity, but this is to be expected with a more complex physical environment.

The opacity-handling routines in the adaptive multigroup algorithm are currently wedded to the IPCRESS data structure. This prevents users from using other popular opacity input formats such as "SESAME." Also, it breaks RAGE's optional analytic opacity routines. The analytic opacities in RAGE are not handled in a discrete way

and thus do not use the same internal data structure. Modifying the existing analytic opacity routines to use an IPCRESS structure, or adding flexibility to the adaptive multigroup opacity handler will be required for these features to work together.

User-defined radiation sources are handled by an existing RAGE routine that is unaware of the existence of a master radiation energy density table. They can be made to work on the coarse radiation energy density table with very little work, but a more elegant solution would be to apply the sources to the master table where the energy dependence can be better modeled. This would require significant changes to the existing RAGE module.

10.1.3 Accuracy improvements

The equal arc-length projection method does put higher energy group resolution in the regions of the representative opacity where there are significant gradients. However, no claim is made that this is an optimal algorithm. This technique was a simple way to arrive at the desired end. There is certainly room for improvement in the intelligence of this method. Pattern matching for K- and L-edges or recognizing regions of tightly packed bound-bound transition lines would lead to better fitting of group boundaries to the major discontinuities in the opacity. Furthermore, fitting a ν^{-3} curve to various portions of the opacity could help find the outer limits of the smooth regions (if any).

The generation of the representative opacity is another area that has significant room for improvement. In particular, the weighting function for choosing how important a given material opacity is to the overall problem ignores any *a priori* user knowledge of what material temperature and density environments will be important to the problem. A certain amount of user input to favorably weight certain environments could help in accurately treating that portion of the problem that the user is particularly interested in.

10.2 Summary

The underlying theory of grey radiation diffusion assumes that it will only be used near strict thermodynamic equilibrium. Data presented in chapter 9 show that for radiation environments that are significantly displaced from thermodynamic equilibrium, grey diffusion gives answers that are both qualitatively and quantitatively invalid.

Correcting these errors requires a method for reintroducing some frequency dependence back into the diffusion equation. The canonical technique for achieving this is static multigroup radiation diffusion. This thesis presents, for the first time, a definitive analysis of shortcomings of static multigroup diffusion when used with a small number of groups.

The outcome of a simulation using static multigroup diffusion is incredibly sensitive to the placement of group boundaries in energy space. Data are presented here that show the accuracy of static multigroup varying wildly as group boundaries are shifted. In some cases, a static multigroup simulation with as many as ten groups is shown to give worse accuracy than the fundamentally inadequate grey diffusion method — despite requiring ten times as long to calculate.

The adaptive multigroup radiation diffusion algorithm described in this thesis and implemented by the author is presented as a cure for this sensitivity. By allowing energy group boundaries to move in time and using a technique for choosing intelligent group boundaries at any given instant, the onus of selecting group boundaries is removed from the user. Furthermore, accuracy of the results shows improvement even over the best-case output for static multigroup diffusion.

The adaptive multigroup algorithm represents a massive improvement over static multigroup diffusion when used on non-equilibrium simulations with small numbers of groups. Data presented here show that the accuracy gained per group is predictable. Also, the accuracy is not sensitive to the energy limits, meaning that the use of multigroup radiation diffusion is now freed of several adjustable parameters or "knobs." The user is no longer required to choose energy limits or group boundaries. The

variance in accuracy caused by the adjustment of these knobs in static multigroup diffusion is alleviated entirely. The result is a vastly simplified system from the user's perspective, as well as a reliable and significant increase in accuracy.

Appendix A

Source Code for Adaptive Multigroup Radiation Diffusion in RAGE

The code module for adaptive multigroup radiation diffusion is divided up into several main subroutines, each of which performs one of the main tasks of the algorithm. These functional blocks include code initialization, choosing group boundaries and creating a coarsened opacity table, remapping changes in the coarse opacity table back into the master opacity table, and handling the master radiation energy density table.

Some minor wrapper functions and support routines have been left out of this appendix for the sake of brevity. The same is true of function headers and other code that does not illustrate how the adaptive multigroup radiation diffusion routine works.

A.1 Module Initialization

The initialization of the adaptive multigroup radiation diffusion module occurs in two steps. The steps are separated by the initialization routines for other RAGE radiation modules, as the output of these modules is needed for further adaptive multigroup

initialization.

The "initial initial" routine simply sets up some data structures for future use and checks the input deck for switches that are now allowed. It also copies the IPCRESS opacity table that was read in the radiation module into the master opacity table data structure.

```
! Do some sanity checks
call mgname('mgrp_num',mgrp_num,.true.)
if(mgrp_num.le.0)then
   ! they have an unreasonable mgrp_num specified
   call global_error('adaptive_multigroup_INITAL0: mgrp_num.le.0')
endif

if(mgrp_num.gt.ipcress_table(4))then
   ! they have specified too many groups
   call global_error('adaptive_multigroup_INITAL0: &
                                    mgrp_num.gt.ipcress_table(4)')
endif

! we are not currently set up to allow user-specified boundary and
! source radiation spectra when using adaptive_multigroup. If the
! user asks for these features, bork.
call mgname('do_usr_bc_spectrum',do_usr_bc,.true.)
call mgname('do_usr_src_spectrum',do_usr_src,.true.)
if(do_usr_bc)then ! they have specified a boundary radiation spectrum
 call global_error('adaptive_multigroup_INITAL0: do_usr_bc_spectrum true')
endif
if(do_usr_src)then ! they have specified a radiation source spectrum
 call global_error('adaptive_multigroup_INITAL0: do_usr_src_spectrum true')
endif
```

```
! Set up the data structure for the coarse groups
coarse_num_groups = mgrp_num
allocate(coarse_group_bnds(coarse_num_groups+1), stat = allostat)
call test_allostat

! Create the master IPCRESS table as a copy of the original
! (this is because the IPCRESS table that is read in from the bliss file
! is the high-res version.  Once adaptive_multigroup_inital is run, the
! ipcress_table variable used by the rest of the RAGE code will actually
! be a coarsened version derived from the master table by this module)
master_table_size = int(ipcress_table(2))
if(allocated(ipcress_table_master))then
   deallocate(ipcress_table_master, stat = allostat) ; call test_allostat
endif
allocate(ipcress_table_master(master_table_size), stat = allostat)
call test_allostat
ipcress_table_master = ipcress_table
```

The second initial routine reads through the master opacity table and generates arrays of the temperature and density points contained within, as well as separating out the opacity information from everything else. It also makes an initial call to the regroup, coarsen, radiation energy density master table initializer, and radiation energy density coarsener. In this way, all of the coarse-mesh data structures are populated correctly with initial data.

```
! Build some data structures based on the contents of the master
! IPCRESS table and populate them with the data extracted from the
! master IPCRESS table.

! IPCRESS table size is:
! 4 +                                   (header)
! num_groups + 1 +                      (group boundaries)
! num_mat * (                           (an entry for every material)
!     4 +                               (material header)
!     num_r +                           (log of density points)
!     num_t +                           (log of temperature points)
!     num_r * num_t +                   (log of gray total opacities)
!     num_r * num_t +                   (log of gray abs. opacities)
!     num_groups * num_t * num_r +      (log of mgrp total opacities)
!     num_groups * num_t * num_r) +     (log of mgrp abs. opacities)
! 1                                     (end of array marker byte)

! NOTE:  Within the multigroup opacities, we store them as num_t
!        groups of contiguous densities.  That is, in
!        "Temperature-Major Order."

! Information about the fine group structure
fine_num_groups = int(ipcress_table_master(4))-1
allocate(fine_group_bnds(fine_num_groups+1), stat = allostat)
```

```
call test_allostat
fine_group_bnds(1:fine_num_groups+1) = &
                          ipcress_table_master(5:fine_num_groups+5)

! Information about the Tempearture and Density grid
j = 8 + fine_num_groups
num_r = ipcress_table_master(j)
num_t = ipcress_table_master(j+1)
allocate(rpoints(num_r), stat = allostat) ; call test_allostat
allocate(tpoints(num_t), stat = allostat) ; call test_allostat
j = j + 2
rpoints(1:num_r) = ipcress_table_master(j:j+num_r-1)
j = j + num_r
tpoints(1:num_t) = ipcress_table_master(j:j+num_t-1)
j = j + num_t

! Information about the gray opacities
phasepoints = num_t * num_r
allocate(tgray(phasepoints), stat = allostat) ; call test_allostat
allocate(agray(phasepoints), stat = allostat) ; call test_allostat
tgray(1:phasepoints) = ipcress_table_master(j:j+phasepoints-1)
j = j + phasepoints
agray(1:phasepoints) = ipcress_table_master(j:j+phasepoints-1)
j = j + phasepoints

! Information about the multigroup opacities
allocate(fine_tmg(phasepoints*fine_num_groups), stat = allostat)
call test_allostat
allocate(fine_amg(phasepoints*fine_num_groups), stat = allostat)
call test_allostat
```

```
allocate(coarse_tmg(phasepoints*coarse_num_groups), stat = allostat)
call test_allostat
allocate(coarse_amg(phasepoints*coarse_num_groups), stat = allostat)
call test_allostat
do g = 1, fine_num_groups
   m = j + (phasepoints * fine_num_groups)
   ! index of start of group g in fine_*mg
   k = ((g-1) * phasepoints) + 1
   fine_tmg(k:k+phasepoints-1) = &
                            ipcress_table_master(j:j+phasepoints-1)
   fine_amg(k:k+phasepoints-1) = &
                            ipcress_table_master(m:m+phasepoints-1)
   j = j + phasepoints
end do ! g     loop over fine groups

! The following two calls are identical to the processing done
! during a normal simulation timestep, they are called here to
! prepare a coarse multigroup structure for the first timestep.

! Determine current desired coarse group boundary placement
call adaptive_multigroup_regroup

! Create a temporary IPCRESS table coursened from the master
call adaptive_multigroup_coarsen

! Create the mgrp_rade_master array and populate it with
! some initial values
call mgrp_rade_master_init

! Populate mgrp_rade with values based on mgrp_rade_master
```

```
call rade_coarsen
```

A.2 The Regroup Routine

This subroutine is where the equal arc-length projection method is implemented. The purpose of this routine is to examine the state of the simulation and select a set of group boundaries for the coarse energy mesh. The number of group boundaries to be chosen is a parameter set in the input deck by the user.

```
! We have a high-resolution IPCRESS table that includes opacities
! for the material at lots and lots of density and temperature
! combinations. This algorithm looks at every cell and grabs its
! (rho,T), then grabs the opacity functions from the IPCRESS table
! for that specific point in (rho,T) space.
!
! Actually, it interpolates between the two nearest Tempearture
! points and the two nearest Density points.  This interpolation is
! done in the same manner as the ipcress_getmu() routine in
! module_ipcress.f90.  No thought was given as to whether or not
! this interpolation could be done in a better way (e.g. bi-rational
! function) - I just wanted to do it the same way as it was done
! elsewhere in the code for the sake of consistency.  At some point,
! this interpolation could be changed but should be changed in both
! places.
!
! The interpolated opacity spectra for each spatial cell in the
! problem are averaged to obtain a "spatially-averaged opacity" for
! the entire problem at the current timestep.  In doing so, it is
! possible that significant structure in the opacity spectra at one
! spatial location might be lost to the averaging process.  However,
! this is acceptable as we need to generate a group structure that
! will work well in all spatial cells, so concentrating on opacity
! features in a given cell's spectra could decrease the
```

```
! applicibility of the resultant group structure elsewhere.
!
! If the opacity structure really changes dramaticaly with space,
! the average opacity will work out to a more-or-less constant
! function and the algorithm degrades gracefully - producing a
! uniform group boundary spacing (in log-energy space) in this case.


! Make some temporary arrays for calculating the average opacity
allocate(sumtmg(fine_num_groups), stat = allostat)
call test_allostat
sumtmg(:) = 0.0_REAL8

m=1  ! FORCE SINGLE-MATERIAL FOR NOW

! Loop over all cells
do n=1,numtop

   l=ltop(n)

   ! Get the log of "partial density" for this material - the
   ! opacity is done by material, so the overall density of a
   ! multi-material cell is not useful.
   density = log(frac_mass(l,m)/frac_vol(l,m))

   ! Find the bounding density indexes from the IPCRESS table.
   rlo = 1
   rhi = num_r
   do while ((rhi-rlo) .gt. 1)
      mid = (rlo+rhi)/2
```

```
   if(density .ge. rpoints(mid)) then
      rlo = mid
   else
      rhi = mid
   endif ! density .ge. rpoints(mid)
enddo ! while
rhi = rlo + 1

! Find the bounding temperature indexes from the IPCRESS table.
tlo = 1
thi = num_t
do while ((thi-tlo) .gt. 1)
   mid = (tlo+thi)/2
   if(log(tev(l)) .ge. tpoints(mid)) then
      tlo = mid
   else
      thi = mid
   endif ! log(tev(l)) .ge. tpoints(mid)
enddo ! while
thi = tlo + 1

! Interpolate the total opacity spectra.
! This routine was basically copied from ipcress_getmu() in
! module_ipcress.f90 and modified to use the master IPCRESS table
! data rather than the dynamic IPCRESS data, as well as changing
! the variable names around some.  The _getmu() routine also only
! retrieved the opacity for a single group, whereas here we are
! doing all groups at once, so some added mechanics had to be
! added for this.
```

```
! Determine where between the two grid points the actual density
! and temperature data lie
sr = (density - rpoints(rlo)) / (rpoints(rhi) - rpoints(rlo))
st = (log(tev(l)) - tpoints(tlo)) / (tpoints(thi) - tpoints(tlo))

! Clip these to a value between zero and one, just to be safe
sruse = max(ZERO, min(ONE, sr))
stuse = max(ZERO, min(ONE, st))

! Loop over groups and perform the interpolation
do f=1, fine_num_groups

   ! Determine the array indexes for the bounding temperatures
   ! and densities
   ! low temperature, low density address
   lolo = ((f-1) * num_r * num_t) + (num_r * tlo) + rlo
   ! low tempearture, high density address
   lohi = lolo + 1
   ! high tempearture, low density address
   hilo = lolo + num_r
   ! high temperature, high density address
   hihi = hilo + 1

   ! Linearly interpolate between the density grid points
   oplo = fine_tmg(lolo) + sruse * (fine_tmg(lohi) - &
                                    fine_tmg(lolo))
   ophi = fine_tmg(hilo) + sruse * (fine_tmg(hihi) - &
                                    fine_tmg(hilo))

   ! Linearly interpolate by temperature between the two points
```

```
      ! calculated above and add to the running sum opacity for
      ! this group.
      sumtmg(f) = sumtmg(f) + oplo + stuse * (ophi-oplo)

   enddo ! f - loop over fine groups
enddo ! n - loop over top-level cells

! Normalize the opacity sum arrays to the number of cells that were
! added together
sumtmg(:) = sumtmg(:) / numtop

! Now sumtmg() contains the "average opacity spectra" for all cells
! at the current timestep.  This compound opacity function is now
! coarsened for use by the rest of the code.

! The first and last boundaries have to match the fine boundary
! structure...
coarse_group_bnds(1) = fine_group_bnds(1)
coarse_group_bnds(coarse_num_groups+1) = &
                                fine_group_bnds(fine_num_groups+1)

! Calculate the "arclength" of the opacity vs. energy spectrum held
! in sumtmg().  We're doing this in log-log space, so we need the
! log of the energy group bounaries as well as the log of the
! opacities in each group:

! Convert the average opacity spectra into log-opacity space
! sumtmg(:) = log(sumtmg(:))

! Calculate the group-center energies (in log-energy space)
```

```
allocate(centerenergy(fine_num_groups), stat = allostat)
call test_allostat
do i=1, fine_num_groups
   centerenergy(i) = (log(fine_group_bnds(i)) + &
                                    log(fine_group_bnds(i+1))) / 2
enddo ! i loop over groups

! Modify the average opacity spectra to accentuate regions we want
! to resolve.
do i=1, fine_num_groups-1
   ! delta energy between centers
   dce = centerenergy(i+1) - centerenergy(i)
   ! delta opacity between groups
   dop = sumtmg(i+1) - sumtmg(i)
   sumtmg(i) = dop/dce
enddo ! i loop over groups (-1)
sumtmg(fine_num_groups) = 0

! Determine the lengths of the lines connecting the group centers
allocate(slopes(fine_num_groups-1), stat = allostat)
call test_allostat
totalarclength = 0.0_REAL8
do i=1, fine_num_groups-1
   ! delta energy between centers
   dce = centerenergy(i+1) - centerenergy(i)
   ! delta opacity between groups
   dop = sumtmg(i+1) - sumtmg(i)
   slopes(i) = dop/dce
   totalarclength = totalarclength + sqrt((dop*dop) + (dce*dce))
enddo ! i loop over groups (-1)
```

```
! Add a segment for the first half of the first group
! distance from left energy boundary to first group center
dx  = centerenergy(1) - log(fine_group_bnds(1))
! slope is rise over run, or dy/dx = slope
dy  = slopes(1) * dx
leftarclength = sqrt((dx*dx) + (dy*dy))
totalarclength = totalarclength + leftarclength

! and a corresponding segment for the last half of the last group
dx  = log(fine_group_bnds(fine_num_groups+1)) - &
                                    centerenergy(fine_num_groups)
! slope is rise over run, or dy/dx = slope
dy  = slopes(fine_num_groups-1) * dx
rightarclength = sqrt((dx*dx) + (dy*dy))
totalarclength = totalarclength + rightarclength

! Determine the desired arclength for the specified number of coarse
! groups
arcsegment = totalarclength / coarse_num_groups

! Define a temporary array to keep track of which fine boundaries
! are already selected
allocate(bndflag(fine_num_groups+1), stat = allostat)
call test_allostat
bndflag(:) = 0
bndflag(1) = 1    ! The first boundary is used by default
totalflags = 1

! Now start at the left energy boundary and walk through the line
```

```
! connecting the group centers.  After each fine group, determine
! if you have exceeded an arcsegment in total length.  If so, flag
! the nearest boundaries for the appropraite number of arcsegments
! that have been accumulated.
!
! The first and last groups have to be treated as special cases
! since they use inner-side slopes on the outer-halves of the
! groups, unlike all interior groups.
totalarclength = 0
do f = 1, fine_num_groups+1
   ! Don't waste time if we've already placed all the boundaries
   ! we need to
   if (totalflags .lt. coarse_num_groups+1) then

      ! calculate the left-half arclength contribution
      if (f .eq. 1) then
         ! handle the special case for the first group, where we
         ! don't have a slopes(0)
         totalarclength = leftarclength
      else
         dx = centerenergy(f) - log(fine_group_bnds(f))
         dy = slopes(f-1) * dx
         totalarclength = totalarclength + sqrt((dx*dx) + (dy*dy))
      endif ! f .eq. 1

      ! calculate the right-half arclength contribution
      if (f .eq. fine_num_groups) then
         ! handle the special case for the last group, where we
         ! don't have a slopes(fine_num_groups)
         totalarclength = totalarclength + rightarclength
```

```
   rightside = rightarclength
else
   dx = log(fine_group_bnds(f+1)) - centerenergy(f)
   dy = slopes(f) * dx
   rightside = sqrt((dx*dx) + (dy*dy))
   ! used later to determine which boundary is nearest
   totalarclength = totalarclength + rightside
endif ! f .eq. fine_num_groups

! Check to see if we have passed an arcsegment in length
if (totalarclength .ge. arcsegment) then

   ! Determine how many arcsegments long the totalarclength
   ! is (truncating to an integer)
   n = INT(totalarclength / arcsegment)

   ! Determine how many of those arcsegments are on the left
   ! and right sides of the group
   left = INT((totalarclength - rightside) / arcsegment)
   right = n - left

   ! Now we can truncate totalarclength for the next iteration
   totalarclength = totalarclength - (n * arcsegment)

   ! Add the appropriate number of flags to each of the
   ! boundaries for this group
   bndflag(f)   = bndflag(f)   + left
   bndflag(f+1) = bndflag(f+1) + right
   totalflags = totalflags + left + right
```

```
      endif ! totalarclength .gt. arcsegment
   endif ! totalflags .lt. coarse_num_groups+1)
enddo ! f loop over fine groups

! Sometimes, due to rounding errors, we can get to this point and
! not have placed the final coarse boundary yet.  So we check for
! that here.
if (totalflags .lt. coarse_num_groups+1) then
   ! Add the missing flags to the last fine boundary
   bndflag(fine_num_groups+1) = bndflag(fine_num_groups+1) + &
                                (coarse_num_groups+1 - totalflags)
endif

! Now we have a list of flagged boundaries in the bndflag() array,
! but some of the boundaries may have been flagged more than once.
! We need to sweep through and make sure this isn't the case.  If
! there is a bndflag(x) > 1, then we need to take the extra
! contribution from this boundary and place it elsewhere.
!
! For now, the algorithm is to start walking out from the heavy
! boundary in both directions until an unflagged boundary is found.
! This boundary is flagged and the heavy bndflag() is decremented.
! This process is repeated until there are no extra flags to
! disperse.
do f=1, fine_num_groups+1
   ! Look for heavy boundaries with more than one flag
   if (bndflag(f) .gt. 1) then

      ! The number of extra flags we need to redistribute
      extr = bndflag(f) - 1
```

```
! Reset the flag to a single usage.
bndflag(f) = 1
left  = f-1
right = f+1
! Start scanning right, slightly favors higher energies
parity = 0

! Keep looking until we've placed all the flags somewhere
do while (extr .gt. 0)

   if (parity .eq. 0) then
      ! Scan right one boundary
      ! Make sure we aren't looking past the right edge of
      ! the array
      if (right .le. fine_num_groups+1) then
         ! See if the boundary is unflagged
         if (bndflag(right) .eq. 0) then
            bndflag(right) = 1    ! Flag it.
            extr = extr - 1       ! That's one down...
            right = right + 1     ! Move the boundary over for
                                  ! the next parity 0 iteration
         else
            right = right + 1
         endif ! bndflag(right) .eq. 0
      endif ! right .le. fine_num_groups+1
      ! Next time scan left.
      parity = 1
   else

      ! Scan left one boundary
```

```
            ! Make sure we aren't looking past the left edge of
            ! the array
            if (left .ge. 1) then
               ! See if the boundary is unflagged
               if (bndflag(left) .eq. 0) then
                  bndflag(left) = 1    ! Flag it.
                  extr = extr - 1      ! That's one down...
                  left = left - 1      ! Move the boundary over for
                                       ! the next parity 1 iteration
               else
                  left = left - 1
               endif ! bndflag(left) .eq. 0
            endif ! left .ge. 1
            ! Next time scan right.
            parity = 0
         endif ! parity .eq. 0|1

         ! Make sure we don't get into a situation where left is on
         ! the global left boundary and right is on the global right
         ! boundary.  If it ever gets there then there is an error,
         ! but it also means that the code will hang in this loop
         ! forever.
         if ((left .eq. 0) .and. (right .eq. fine_num_groups+2))then
            call global_error('adaptive_multigroup_REGROUP: &
                                       no open boundaries to flag!')
         endif

      enddo ! while extr .gt. 0
   endif ! bndflag(f) .gt. 1
enddo ! f loop over fine groups
```

```
! At this point, the bndflag() array should contain
! coarse_num_groups+1 elements equal to 1 and the remainder equal
! to 0.  The following loop tests this assertion and builds the
! coarse_group_bnds() array from the results.
index = 1;
do f=1, fine_num_groups+1
   if (bndflag(f) .eq. 1) then
      ! Copy this fine group boundary energy over for use as a
      ! coarse group boundary
      coarse_group_bnds(index) = fine_group_bnds(f)
      index = index + 1
   elseif (bndflag(f) .gt. 1) then
      ! There shouldn't be any flags with counts greater than 1
      ! after the above do loop.
      call global_error('adaptive_multigroup_REGROUP: &
                                          bndflag(f) .gt. 1')
   endif ! bndflag(f) .eq. 1
enddo ! f loop over fine group boundaries

! Just a sanity check, index should now be equal to
! coarse_num_groups + 1
if (index .ne. (coarse_num_groups+2)) then
   call global_error('adaptive_multigroup_REGROUP: &
                               index .ne. coarse_num_groups+2')
endif

! Memory cleanup
deallocate(sumtmg)
deallocate(centerenergy)
```

A.3 The Coarsen Routine

This subroutine takes the high-resolution master opacity data and coarsens it down onto the low-resolution mesh defined in the regroup routine. It then builds a properly-formatted IPCRESS table with this coarse data and puts it where the rest of RAGE will be expecting to find the (usually) static opacity data.

```
! Sanity check the correspondance between the coarse and fine energy
! grids.  If low boundaries dont match or high boundaries dont
! match, then bork.

if(abs((fine_group_bnds(1) - coarse_group_bnds(1))) .gt. &
                    1.0E-14_REAL8)then ! left grid edges dont match
   call global_error(adaptive_multigroup_COARSEN: lower energy &
                                                 limits do not match)
endif
if(abs((fine_group_bnds(fine_num_groups+1) - &
      coarse_group_bnds(coarse_num_groups+1))).gt.1.0E-14_REAL8)then
   call global_error(adaptive_multigroup_COARSEN: upper energy &
                                                 limits do not match)
endif

! Calculate the multigroup opacities for each of the coarse groups.
! This involves taking a Rosseland mean of the constituent fine
! groups.

fine_grp_upper = 0

do g = 1, coarse_num_groups

   coarse_bnd_lower = coarse_group_bnds(g)
```

```
fine_grp_lower = fine_grp_upper + 1
fine_bnd_lower = fine_group_bnds(fine_grp_lower)

! Sanity check that the current coarse lower boundary corresponds
! to the current fine lower boundary.
if((coarse_bnd_lower - fine_bnd_lower) .gt. 1.0E-14_REAL8)then
   ! left boundaries dont match!
   call global_error(adaptive_multigroup_COARSEN: &
                              coarse_bnd_lower.ne.fine_bnd_lower)
endif

! find the fine boundary that matches the right coarse boundary.
coarse_bnd_upper = coarse_group_bnds(g+1)
fine_bnd_upper = fine_group_bnds(fine_grp_upper+1)
do while (fine_bnd_upper.lt.coarse_bnd_upper)
   fine_grp_upper = fine_grp_upper + 1
   fine_bnd_upper = fine_group_bnds(fine_grp_upper+1)
enddo ! while

! Sanity check that the upper boundaries are now the same
if((coarse_bnd_upper - fine_bnd_upper) .gt. 1.0E-14_REAL8)then
   ! right boundaries dont match!
   call global_error(adaptive_multigroup_COARSEN:  &
                           coarse_bnd_upper.ne.fine_bnd_upper)
endif

! now we have a series of fine energy bins from fine_grp_lower
! to fine_grp_upper that exactly span the energy space of
! coarse group g.
```

```
! CALCULATE THE INTEGRAL OF dB/dT OVER THE COARSE ENERGY RANGE
! SUM THE NUMERATOR TERMS OF THE CONSTITUENTS FINE GROUPS
! coarse_amg = INTEGRAL/SUM (absorption)
! coarse_tmg = INTEGRAL/SUM (total)

do t_index = 1, num_t
   t = tpoints(t_index)
   do r_index = 1, num_r

      numerator_tmg = 0
      numerator_amg = 0

      ! Handle the trivial case where there is only one fine
      ! group spanning the coarse group.  Just copy the fine
      ! group opacities over to the coarse group array.
      if (fine_grp_lower .eq. fine_grp_upper) then
         coarse_tmg((g-1) * (num_r * num_t) + ((t_index-1) * &
             num_r) + r_index) = fine_tmg((fine_grp_lower-1) * &
             (num_r*num_t) + ((t_index-1)*num_r) + r_index)
         coarse_amg((g-1) * (num_r * num_t) + ((t_index-1) * &
             num_r) + r_index) = fine_amg((fine_grp_lower-1) * &
             (num_r*num_t) + ((t_index-1)*num_r) + r_index)
      else

         ! There are multiple fine groups spanning the coarse
         ! group.
         do f = fine_grp_lower, fine_grp_upper

            ! CALCULATE THE NUMERATOR AND DENOMINATOR COMPONENTS
            ! OF THE ROSSELAND MEAN FOR GROUP f
```

```
!
! The master IPCRESS table holds the log of the
! values for the Rosseland mean opacities (total and
! absorption) for each of the fine cells.  We
! separate the definition of the inverse Rosseland
! mean into its numerator and denominator components
! such that the denominator is just the integral
! across the group energy range of the derivative of
! the Planck function with respect to temperature.
! By calculating the value of this integral, we can
! multiply it by the inverse of the opacity to obtain
! a value for the numerator.
!
! These numerators can be added for each of the
! constituent fine groups, f, which divide the coarse
! group, g.  When this sum is computed, the total
! denominator for the entire energy span of coarse
! group is calculated in the same manner as above.
! This denominator is used to divide the total
! numerator calculated above to reveal the value of
! the inverse of the Rosseland mean for the coarse
! group.  The log of this value is stored in the
! coarse IPCRESS table for use by the rest of the
! code.

! retrieve the log(opacities) for this fine group,
! density, and temperature
tmg = fine_tmg((f-1) * (num_r * num_t) + &
           ((t_index-1)*num_r) + r_index)
amg = fine_amg((f-1) * (num_r * num_t) + &
```

```
            ((t_index-1)*num_r) + r_index)

! convert from log(opacity) to opacity
tmg = exp(tmg)
amg = exp(amg)

! get the inverse of the opacities
tmg = 1 / tmg
amg = 1 / amg

! Calculate dimensionless energies for integration
! limits
!
! The routine rosseland_integral expects a pair of
! integration limits - dimensionless energies defined
! as:
!
!    u = hv/kT
!
! However, the temperatures stored in our master
! IPCRESS table are actually the natural logs of kT
! in units of eV:
!
!    t = ln(kT(in eV))
!
! The group boundary energies stored in the master
! IPCRESS table are in units of keV so to come up
! with hv/kT and make it dimensionless we make the
! following substitutions:
!
```

```
!    kT = exp(t)                          in eV
!    hv = 1000 * fine_group_bnds(x)       in eV
!
! Now we can write our dimensionless energy
! integration boundaries as:
!
!    u = 1000 * exp(-t) * fine_group_bnds(x)
!

u1 = fine_group_bnds(f) * exp(-t) * 1000.0E0_REAL8
u2 = fine_group_bnds(f+1) * exp(-t) * 1000.0E0_REAL8
call rosseland_integral(u1, u2, denominator)

! To make the integral solved by rosseland_integral
! equivalent to the integral of dB/dT with respect to
! frequency, we could multiply the result by a factor
! of:
!
!    (2 * k^4 * T^3) / (c^2 * h^3)
!
! But again we can replace a term of (kT)^3 with
! (exp(t))^3 == exp(3t), leaving:
!
!    exp(3t) *  (2 * k) / (c^2 * h^3)
!
! where      k = 1.3806503E-23 m^2kg/(s^2K)
!            c = 2.99792458E8  m/s
!            h = 6.626068E-34  m^2kg/s
!
!    2k/(c^2 * h^3) = 1.05609896E60 s^3/(K kg^2 m^6)
```

```
!
! However, since were expressing kT in units of eV,
! we need to convert the remainder of this term to
! the correct energy units as well.
!
!    1 K = 8.617343E-5 eV
!
!    2k/(c^2 * h^3) = 1.22555057E64 s^3/(eV kg^2 m^6)
!
!
! denominator = denominator * 1.22555057E64_REAL8 *
!               exp(3.0E0_REAL8 * t)
!
! But since were just going to reconstruct one of
! these denominators again later and divide out by
! it, this constant cancels anyway.
!
! If the denominator returned was zero, then the
! hv/kT value was in a regime that receives such a
! low Rosseland weight that the integration routine
! cant even calculate a number that small.  In that
! case, well just ignore this groups contribution
! to the Rosseland mean, as it is guaranteed to be
! amazingly small.

if (denominator .gt. 0.0E0_REAL8) then
   ! Otherwise, we compute the numerators from the
   ! known opacities and denominators and add it to
   ! the running sum of numerators for all fine
   ! groups spanning group g.
```

```
      numerator_tmg = numerator_tmg +(tmg * denominator)
      numerator_amg = numerator_amg +(amg * denominator)
   endif

end do ! f - loop over fine groups within g

! At this point, it is still possible that we have a
! zero numerator.  If all of the fine groups spanning
! coarse group g returned a zero for their denominator
! integral, then none of them would have added anything
! to the numerator sums and we are left with nothing
! with which to construct a valid group opacity.
! Instead, we need to go back and calculate the
! numerator contributions in a different way.

if ((numerator_tmg .eq. 0.0E0_REAL8) .or. &
    (numerator_amg .eq. 0.0E0_REAL8)) then

   numerator_tmg = 0.0E0_REAL8
   numerator_amg = 0.0E0_REAL8

   ! We want to multiply the numerator and denominator
   ! integrands in the inverse Rosseland mean formula by
   ! some factor that will make at least the largest of
   ! the very small Rosseland weights computationally
   ! tractable.  For large values of the dimensionless
   ! energy, u, the Rosseland integral integrand rapidly
   ! approaches exp(-u).  Thus, the smaller the value of
   ! u, the larger the group weight will be. Because
   ! u = hv/kT and everything is effectively constant
```

```
! except v, we can restate this as "The first fine
! energy group within coarse group g,
! f=fine_grp_lower, is the most significant and will
! have the largest Rosseland weight."
!
! To normalize the calculation to the weight of the
! most-significant fine group, we take the lowest
! energy in coarse group g, convert it to a
! dimensionless energy, and call this value U.

U = coarse_group_bnds(g) * exp(-t) * 1000.0E0_REAL8

! Now we change the equation for the Rosseland mean
! such that the integrands are tractable at this
! energy.  We do this by multiplying top and bottom
! by exp(U).  Thus, the high-u limit formula for the
! inverse Rosseland mean becomes:
!
! 1/Kr = exp(U)*integral{u1->u2} [exp(-u)/Kv du] /
!                exp(U)*integral{u1->u2} [exp(-u) du]
!      = exp(U)*integral{u1->u2} [exp(-u)/Kv du] /
!                 integral{u1->u2} [exp(-u)exp(U) du]
!      = exp(U)*integral{u1->u2} [exp(-u)/Kv du] /
!                integral{u1->u2} [exp(U-u) du]
!
! The value of U is constant regardless of which fine
! group is being treated, so the exp(U) term is a
! constant and can be pulled into the integral as
! show above.  The equation has been multiplied by
! exp(U)/exp(U) = 1 and should therefore still
```

```
! produce the same result.  However, instead of
! having to deal with exp(-u), which can be too small
! to be handled by the code, we deal with exp(U-u).
!
! For all energies in group g, u >= U so the value of
! exp(U-u) will be <= 1.  However, at the low-energy
! end of the first fine group within coarse group g,
! the the value of exp(U-u) will be approximately
! unity.  In this way, we ensure that at least one
! fine group will return a non-zero denominator.

do f = fine_grp_lower, fine_grp_upper

   u1 = fine_group_bnds(f) * exp(-t) * 1000.0E0_REAL8
   u2 = fine_group_bnds(f+1) * exp(-t) * &
                                      1000.0E0_REAL8

   ! There is no need to call rosseland_integral()
   ! for calculating these new denomninators, as the
   ! integral of exp(U-u) has an analytical solution:
   !
   ! integral{u1->u2} [exp(U-u) du] = exp(U-u1) -
   !                                  exp(U-u2)

   denominator = exp(U-u1) - exp(U-u2)

   ! Some of the fine groups might still return zero
   ! denominators - just not the first one.  We still
   ! need to check whether or not to ignore them.
```

```
   if (denominator .gt. 0.0E0_REAL8) then
      ! Compute the numerators from the known
      ! opacities and denominators and add it to the
      ! running sum of numerators for all fine groups
      ! spanning group g.
      numerator_tmg = numerator_tmg + (tmg * &
                                       denominator)
      numerator_amg = numerator_amg + (amg * &
                                       denominator)
   endif

end do ! f - loop over fine groups within g

! All that remains is to ensure that we calculate the
! denominator for the coarse group in the same way
! and with the same constant U.

u1 = coarse_group_bnds(g) * exp(-t)   *1000.0E0_REAL8
u2 = coarse_group_bnds(g+1) * exp(-t) *1000.0E0_REAL8
denominator = exp(U-u1) - exp(U-u2)

! Knowing the sum of the fine group opacity
! numerators, calculated inside the f loop above, and
! the coarse group denominator just obtained, we can
! reconstruct the Rosseland mean opacities for the
! coarse group. The log of the result is taken in
! order to be consistent with module_ipcress
! handling of IPCRESS opacities.
coarse_tmg((g-1) * (num_r * num_t) + ((t_index-1) * &
   num_r) + r_index) = log(denominator/numerator_tmg)
```

```
            coarse_amg((g-1) * (num_r * num_t) + ((t_index-1) * &
               num_r) + r_index) = log(denominator/numerator_amg)

         else  ! neither of the numerator sums were zero

            ! Now were back in the case where we got good values
            ! for the denominators using rosseland_integral() so
            ! we need to reconstruct the denominator for the
            ! coarse group g in the same way.
            u1 = coarse_group_bnds(g) * exp(-t) * 1000.0E0_REAL8
            u2 = coarse_group_bnds(g+1) * exp(-t) * &
                                                  1000.0E0_REAL8
            call rosseland_integral(u1, u2, denominator)

            ! Getting the coarse group opacities is now handled
            ! the same way as before, though with the original
            ! numerator value and the rosseland_integral version
            ! of the denominator. The log of the result is stored
            ! to be consistent with module_ipcress treatment of
            ! IPCRESS opacities.
            coarse_tmg((g-1) * (num_r * num_t) + ((t_index-1) * &
               num_r) + r_index) = log(denominator/numerator_tmg)
            coarse_amg((g-1) * (num_r * num_t) + ((t_index-1) * &
               num_r) + r_index) = log(denominator/numerator_amg)

         endif ! - valid numerator sum not generated using
               !   rosseland_integral()
      endif !     - multiple fine group case
   end do ! r  - loop over density points
end do ! t    - loop over temperature points
```

```
end do ! g  - loop over coarse groups

! at this point we have collected everything we need to produce a
! RAGE-formatted IPCRESS table that incorporates the coarse group
! structure.  All that is left is to create and populate the
! actual data structure.

! CREATE THE NEW COARSE-GROUPED IPCRESS TABLE
! for array structure info, see ipcress_load() in module_ipcress.f90

! start by nuking the old table
if(allocated(ipcress_table))then
   deallocate(ipcress_table, stat = allostat) ; call test_allostat
endif

! Create a new table array
!
! table size is:
! 4 +                          (header)
! coarse_num_groups + 1 +      (group boundaries)
! num_mat * (                  (an entry for every material)
!    4 +                       (material header)
!    num_r +                   (density points)
!    num_t +                   (temperature points)
!    num_r * num_t +           (gray total opacities)
!    num_r * num_t +           (gray absorption opacities)
!    coarse_num_groups * num_r * num_t +  (mgrp total opacities)
!    course_num_groups * num_r * num_t) + (mgrp abs. opacities)
! 1                            (a null entry at the end)
```

```
num_mat = 1
table_size = 6 + coarse_num_groups
mat_record_size = 4 + num_r + num_t + 2 * ((1 + coarse_num_groups) &
                                          * (num_r * num_t))
table_size = table_size + (num_mat * mat_record_size)
allocate(ipcress_table(table_size), stat = allostat)
call test_allostat

! fill in the header and the null footer
ipcress_table(1) = 1
ipcress_table(2) = table_size
ipcress_table(3) = coarse_num_groups + 6
ipcress_table(4) = coarse_num_groups + 1
ipcress_table(table_size) = 0

! load in the new group boundaries
i = 5 + coarse_num_groups
ipcress_table(5:i) = coarse_group_bnds(1:coarse_num_groups+1)
i = i + 1

! loop over materials  (currently not implemented)
!  do m = 1,nmat

! fill in the material header
ipcress_table(i) = i+mat_record_size
ipcress_table(i+1) = ipcress_table_master(7+fine_num_groups)
ipcress_table(i+2) = num_r
ipcress_table(i+3) = num_t
i = i + 4
```

```
! fill in the density points
ipcress_table(i:i+num_r-1) = rpoints(1:num_r)
i = i + num_r

! fill in the tempearture points
ipcress_table(i:i+num_t-1) = tpoints(1:num_t)
i = i + num_t

! fill in the gray total opacities
j = i + (num_t * num_r) - 1
ipcress_table(i:j) = tgray
i = j + 1

! fill in the gray absorption opacities
j = i + (num_t * num_r) - 1
ipcress_table(i:j) = agray
i = j + 1

! fill in the multigroup total opacities
m = num_t * num_r
do g = 1,coarse_num_groups
   j = i + m - 1
   k = ((g - 1) * m) + 1
   ipcress_table(i:j) = coarse_tmg(k:k+m-1)
   i = j + 1
end do ! g    groups loop

! fill in the multigroup absorption opacities
do g = 1,coarse_num_groups
   j = i + m - 1
```

```
   k = ((g - 1) * m) + 1
      ipcress_table(i:j) = coarse_amg(k:k+m-1)
   i = j + 1
end do ! g    groups loop
```

A.4 Initialization of the Radiation Energy Density Master Table

This piece of code fills the master radiation energy density table with a Plankian distribution for the local material temperature in each cell. It consists of two parts - a wrapper function and a method for calculating the Planck weight of each group.

The wrapper function

```
! create the master and backup mgrp_rade data structures
if(allocated(mgrp_rade_master))deallocate(mgrp_rade_master)
allocate(mgrp_rade_master(numtop,fine_num_groups))
if(allocated(mgrp_rade_old))deallocate(mgrp_rade_old)
allocate(mgrp_rade_old(numtop,coarse_num_groups))

! go through each cell and raise any really-low rade values to some
! minimum value then create the rev() array where
! rev(x) = (rade(x) * arad)**(1/4)
! NOTE: module multigroup will re-do this when it's inital() is
! called, but we need the values now so we'll just beat them to it.
do n=1,numtop
   l=ltop(n)
   rade(l) = max(rademn, rade(l))
   rev(l) = sqrt(sqrt(rade(l)/A_RAD))
enddo ! n

! For each fine group, calculate an array of Planck weights for each
! cell in the problem. Add this data to the mgrp_rade_master data
! strucure.
do g=1,fine_num_groups
   call adaptive_multigroup_get_plnk_wt(g,rev,plnk_wt)
```

```
   do n=1,numtop
      l=ltop(n)
      mgrp_rade_master(l,g)=plnk_wt(l)*rade(l)
   enddo ! n
enddo ! g

! Replicate the call to multigroup_movedn() found in
! multigroup_inital_planck()
do g=1,fine_num_groups
   call movdnw(mgrp_rade_master(:,g),vcell)
enddo ! g
```

The Planck weight calculator

```
! Initialize the Planck weights to their maximum value for all
! cells.
plnk_wt(1:numcell)=ONE

! If there is only one fine group [weird but possible] then the
! solution is trivial and we don't have to do any work.
if(fine_num_groups.gt.1)then

   ! Loop over all cells and construct integration limit arrays for
   ! the Planck integral. Dimensionless energy u = hnu/kT where the
   ! photon energies are stored in the group boundaries and the
   ! material energies are passed in through to use().
   do n=1,numtop
      l=ltop(n)
      u1(n)=fine_group_bnds(group)/tev_use(l)
      u2(n)=fine_group_bnds(group+1)/tev_use(l)
   enddo ! n
```

```
   ! The planck_integral() function returns an integral from u to
   ! infinity, so to get the integral from u1 to u2, we have to make
   ! two calls and subtract the results.
   call planck_integral(numtop,u1,integral1)
   call planck_integral(numtop,u2,integral2)

   ! Loop over all cells again and calculate the Planck weight based
   ! on the results of the Planck integral calls.
   do n=1,numtop
      l=ltop(n)
      exp12=exp(u1(n)-u2(n))
      weight=exp(-u1(n))*(integral1(n)-exp12*integral2(n))
      plnk_wt(l)=min(ONE,max(ZERO,weight))
   enddo ! n

endif
```

A.5 Coarsening of the Radiation Energy Density

This routine creates a multigroup radiation energy density array with a value for each cell and each coarse energy group. The source data is the master radiation energy density table. A backup copy is made for comparison after the timestep has run, allowing for calculations of the changes in radiation energy density over the coarse of a timestep.

```
! Create the temporary array for the group weights for mgrp_rade
if(allocated(weight))then
   call global_error('adaptive_multigroup_RADE_COARSEN: weight &
                                                 already allocated')
endif
allocate(weight(numcell, coarse_num_groups), stat=allostat)
call test_allostat
weight(:,:) = 0.0_REAL8

! Initalize the array of coarse group weights
do g=1,coarse_num_groups
   do n=1,numtop
      l=ltop(n)
      weight(l,g) = 0
   enddo ! n
enddo ! f

! Renormalize mgrp_rade_master() to ensure that the sum over groups
! is equal to rade in each cell.  This call replicates the
! functionality of mgmfg_renorm() in module_radiation_mgmfg.f90
! except that it operates on the rade master as well.

do n = 1,numtop
```

```
   l = ltop(n)
   sum = ZERO
   do f = 1,fine_num_groups
      sum = sum + mgrp_rade_master(l,f)
   enddo ! f
   if(sum.gt.ZERO)then
      do f = 1,fine_num_groups
         mgrp_rade_master(l,f) = rade(l) * mgrp_rade_master(l,f)/sum
      enddo
   endif
enddo ! n


! Loop over all fine groups and calculate the coarse weights, which
! are the sum of the weights of the constituent fine groups.  The
! weight for a group and cell is defined as the multigroup
! radiation energy density for that group and cell, divided by the
! grey radiation energy density for that cell.
!
! weight(g,l) = mgrp_rade(g,l) / rade(l)
!

g = 1
do f=1,fine_num_groups
   ! Check to see if we have passed the far end of the current
   ! coarse group
   if(fine_group_bnds(f) .ge. coarse_group_bnds(g+1))then
      g = g + 1
   endif
   do n=1,numtop
```

```
      l=ltop(n)
      ! add the fine group's weight to the coarse group's
      ! cummulative weight.
      weight(l,g) =(mgrp_rade_master(l,f) / rade(l)) + weight(l,g)
   enddo ! n
enddo ! f

! Update the coarse multigroup radiation energy density based on the
! weights that we just calculated in the above loop.  Also create a
! backup copy of the mgrp_rade for use later in comparing the before
! and after values to retrieve the deltas imposed by the upcoming
! timestep.
do g=1,coarse_num_groups
   do n=1,numtop
      l=ltop(n)
      mgrp_rade(l,g) = weight(l,g) * rade(l)
      mgrp_rade_old(l,g) = mgrp_rade(l,g)
   enddo ! n
enddo ! g

! Renormalize the new coarse mgrp_rade() to make sure the sum of its
! values is equal to the grey rade.
call mgmfg_renorm

! Free up some memory - these data structures are big!
deallocate(weight)
```

A.6 Refining the changes in Radiation Energy Density

This routine calculates the changes in radiation energy density that occurred over the previous timestep and remaps those changes back into the high-resolution master table for radiation energy density via a piecewise linear interpolation routine. This subroutine is called at the end of each timestep, after all routines that might alter the coarse radiation energy density have run.

```
! Create the temporary array for the temporal changes in mgrp_rade
if(allocated(delta_mgrp_rade))then
   call global_error('adaptive_multigroup_RADE_COARSEN:  &
                              delta_mgrp_rade already allocated')
endif
allocate(delta_mgrp_rade(numtop, coarse_num_groups), stat=allostat)
call test_allostat

! Calculate the temporal change in mgrp_rade over the last timestep
delta_mgrp_rade(:,:) = mgrp_rade(:,:) - mgrp_rade_old(:,:)

! Now the tricky part - we need to map these changes in mgrp_rade
! back onto the master array, mgrp_rade_master.  This means coming
! up with some distribution of the change in coarse group g between
! the constituent fine groups {f}.  A first-order solution would be
! to just apply the coarse delta to each of the fine cells.  A
! higher-order solution is the piecewise linear fit.  Even fancier
! would be to do some cubic spline fits, integrate with gauss
! quadriture points, etc.  For now I'm just implementing a simple
! van Leer piecewise linear fit.

! The first thing to do is to create an array of the midpoints of
```

```
! each coarse energy group.
do g=1, coarse_num_groups
   center(g) = (log10(coarse_group_bnds(g)) +  &
                log10(coarse_group_bnds(g+1)))/2
enddo ! g

do n=1,numtop
   l=ltop(n)

   ! The slope of the line connecting the group centers is dy/dx
   ! where  dy = change in delta_mgrp_rade between the two cells
   ! and dx = distance between cell centers
   do g=1,coarse_num_groups-1
      dx = center(g+1) - center(g)
      slope(g) = (delta_mgrp_rade(l,g+1) - delta_mgrp_rade(l,g))/dx
   enddo ! g

   f = 1
   do g=1,coarse_num_groups
      if(g.eq.1)then
         ! Special case for group 1, where a low-side slope was not
         ! possible.  Just use the high-side slope as it is the only
         ! one available.
         m = slope(g)
      elseif(g.eq.coarse_num_groups)then
         ! Special case for the last group, where a high-side slope
         ! was not possible. Just use the low-side slope as it is
         ! the only one available.
         m = slope(coarse_num_groups-1)
      else
```

```
   ! Determine whether the low-side or high-side slope has the
   ! lower magnitude.
   if(abs(slope(g-1)) .lt. abs(slope(g)))then
      m = slope(g-1)
   else
      m = slope(g)
   endif
endif

! We now have the three pieces of information we need to
! determine the relative contribution to the change in
! multigroup radiation energy density for any fine group.
! These three things are: the center energy for the coarse
! group containing the fine group (C), the slope of the
! piecewise linear fit for the coarse group containing the
! fine group (m), and the total change in multigroup radiation
! energy density for the coarse group (d).
!
! We can write the equation for the piecewise linear fit as
! y = mx+b where
!
! y  is the change in mgrp_rade
! m  is the slope, selected above
! x  is the photon energy
! b  is the "y-intercept", i.e. the value of the fit for
!       photon energy = 0
!
! The y-intercept can be calculated from the slope and the
! center energy of the group, and comes out to be:
!
```

```
! b = d - m*C
!
! So we can re-write the equation of the fit as
!
! y = mx - mC + d
!
! As stated above, we now know m, C, and d for each coarse
! group so we have one equation and one unknown and can find y
! at any point x.  However, what we want is the total change
! to be attributed over the width of each fine group, so we
! need to calculate the area under this line.  For this, we
! integrate the fit line equation to get:
!
! Area = x_2*(0.5*m*x_2 - m*C + d) - x_1*(0.5*m*x_1 - m*C + d)
!
! where x_1 is the energy of the left fine group boundary and
! x_2 is the right boundary.  Now, all we have to do is solve
! this equation for each fine group within group g and add the
! result to the proper spot in the master multigroup
! radiation energy density array.
!
! The slope, m, is calculated in log10 energy space, so we
! need to make sure that we are using log10 energies for x_1,
! x_2, and C as well.   The center energies were already
! calculated in log space, above, so all we have to do here is
! put log10() modifiers arund the calls to fine_group_bnds()
! for retrieving the values of x_1 and x_2.

! Loop over all fine groups within coarse group g
do while((f.le.fine_num_groups) .and. (fine_group_bnds(f) &
```

```
                                    .lt. coarse_group_bnds(g+1)))
          ! Calculate the two terms in the area under the linear fit
          term1 = 0.5_REAL8 * m * log10(fine_group_bnds(f+1)) -  &
                                  m * center(g) + delta_mgrp_rade(l,g)
          term1 = term1 * log10(fine_group_bnds(f+1))
          term2 = 0.5_REAL8 * m * log10(fine_group_bnds(f)) -  &
                                  m * center(g) + delta_mgrp_rade(l,g)
          term2 = term2 * log10(fine_group_bnds(f))

          ! Area = term1 - term2; we add this area to the master rade
          ! table
          mgrp_rade_master(l,f) = max(ZERO, mgrp_rade_master(l,f) &
                                                  + term1 - term2)

          ! and we move on to the next fine group.
          f = f + 1
       enddo ! while f in g
    enddo ! g  - loop over all coarse groups
enddo ! n     - loop over all cells

! Free up some memory
deallocate(delta_mgrp_rade)
```

Bibliography

[1] Thomas Betlach. Multifrequency-gray method for multigroup radiation diffusion. Technical report, SAIC, San Diego, CA, April 1999.

[2] P.A. Bradley and D.C. Wilson. Physics of one-dimensional capsule designs for the national ignition facility. *Physics of Plasma*, 6(11):4293–4303, November 1999.

[3] J. Robert Buchler and William R. Yueh. Compton scattering opacities in a partially degenerate electron plasma at high temperatures. *Astrophysical Journal*, 210:440–446, 1976.

[4] S. Chandrasekhar. *Radiative Transfer.* Dover, Mineola, New York, second edition, 1960.

[5] Karl-Heinz A. Winkler Dimitri Mihalas and Michael L. Norman. Adaptive-mesh radiation hydrodynamics - ii. the radiation and fluid equations in relativistic flows. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 31(6):479–489, 1984.

[6] E. A. Dorfi and L. O'C. Drury. Simple adaptive grids for 1-d initial value problems. *Journal of Computational Physics*, 69(1):175–195, March 1987.

[7] E.A. Dorfi and A. Gautschy. Simple adaptive grids fpr astrophysical problems. In J.R. Buchler, editor, *The Numerical Modelling of Nonlinear Stellar Pulsations*, pages 289–302. Kluwer Academic Publishers, The Netherlands, September 1990.

[8] H. Dreicer. Kinetic theory of an electron-photon gas. *Physics of Fluids*, 7:735–753, 1964.

[9] M. Gehmeyr and D. Mihalas. Adaptive grid radiation hydrodynamics with titan. *Physica D*, 77:320–341, 1994.

[10] Edward G. Harris. *A Pedestrian Approach to Quantum Field Theory.* John Wiley & Sons, Inc., New York, NY, 1972.

[11] W. Heitler. *The Quantum Theory of Radiation.* Oxford University Press, Oxford, 1954.

[12] W. Kass and M. O'Keeffe. Numerical solution of fick's equation with concentration-dependent diffusion coefficients. *Journal of Applied Physics*, 37:2377, 1966.

[13] Randall J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhauser, Boston, 1992.

[14] C. D. Levermore. UCID-18229: A chapman-enskog approach to flux-limited diffusion theory. Technical report, Lawrence Livermore Laboratory, Livermore, CA, 1979.

[15] C. D. Levermore. UCRL-87550: Relating eddington factors to flux limiters. Technical report, Lawrence Livermore Laboratory, Livermore, CA, December 1981.

[16] C. D. Levermore. UCID-19361: Bounds on eddington factors. Technical report, Lawrence Livermore Laboratory, Livermore, CA, January 1982.

[17] C. D. Levermore and G. C. Pomraning. A flux-limited diffusion theory. *The Astrophysical Journal*, 248:321–334, August 1981.

[18] John D. Lindl. Development of the indirect-drive approach to inertial confinement fusion and the target physics basis for ignition and gain. *Physics of Plasma*, 2(11):3933–4024, November 1995.

[19] John D. Lindl. *Inertial Confinement Fusion.* Springer-Verlag, New York, NY, 1998.

[20] C. M. Lund and J.R. Wilson. UCRL-84678: Some numerical methods for time dependant multifrequency radiation transport calculations in one dimension. Technical report, Lawrence Livermore National Laboratory, Livermore, CA, 1980.

[21] R.E. Marshak. Effect of radiation on shock wave behavior. *Physics of Fluids*, 1:24–29, 1958.

[22] Dimitri Mihalas. *Stellar Atmospheres.* W.H.Freeman and Company, San Francisco, California, second edition, 1978.

[23] Dimitri Mihalas and Larry H. Auer. An X-6 radiation hydrodynamics primer. Technical report, Los Alamos National Laboratory, Los Alamos, NM, February 2000.

[24] Dimitri Mihalas and Barbara Weibel-Mihalas. *Foundations of Radiation Hydrodynamics.* Dover, Mineola, New York, second edition, 1999.

[25] Isaac Newton. *Philosophiae naturalis principia mathematica.* Joseph Streater for the Royal Society, London, UK, 1687.

[26] Albert G. Petschek and Ralph E. Williamson. LAMS-2421: The penetration of radiation with constant driving temperature. Technical report, Los Alamos Scientific Laboratory, Los Alamos, NM, 1960.

[27] G. C. Pomraning. GA-9122: The extended eddington approximation with anisotropic scattering. Technical report, Gulf General Atomic, Inc., San Diego, CA, December 1969.

[28] G. C. Pomraning. A modification to the conventional equation of transfer needed to insure consistency with the equilibrium diffusion description. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 14:657–660, 1974.

[29] G. C. Pomraning. Maximum entropy eddington factors and flux limited diffusion theory. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 26(5):385–388, 1981.

[30] G. C. Pomraning. Flux limiters and eddington factors. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 27(5):517–530, 1982.

[31] G. C. Pomraning. LA-UR-82-2625: Radiation hydrodynamics. Technical report, Los Alamos National Laboratory, Los Alamos, NM, 1982.

[32] G.C. Pomraning. A moments method for describing the diffusion of radiation from a cavity. *Journal of Applied Physics*, 38:3845, 1967.

[33] G.C. Pomraning. A moments method for describing the diffusion of radiation from a cavity ii. *Journal of Applied Physics*, 39:1479, 1968.

[34] G.C. Pomraning. Propagation of a non-spherical thermal wave. *Journal of Applied Physics*, 43:2722, 1972.

[35] William H. Press. *Numerical Recipes in C.* Cambridge University Press, Cambridge, UK, 1999.

[36] Dina Prialnik. *Stellar Structure and Evolution.* Cambridge University Press, New York, NY, 2000.

[37] Robert D. Richtmyer and K.W. Morton. *Difference Methods for Initial-Value Problems.* Krieger, Malabar, Florida, 1967.

[38] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, Boston, MA, 1996.

[39] Douglas H. Sampson. The opacity at high tempeartures due to compton scattering. *Astrophysical Journal*, 129:734–751, 1959.

[40] Christina Scovel. LA-UR-04-2959: Sage users manual. Technical report, Los Alamos National Laboratory, Los Alamos, NM, 2004.

[41] Mikhail Shashkov. *Conservative Finite-Difference Methods on General Grids.* CRC Press, Boca Raton, Florida, 1996.

[42] R. Nelson Byrne Tom Betlach and Michael Gittings. LA-UR-98-5677: Radiation diffusion in RAGE. Technical report, LANL and SAIC, Los Alamos, NM, October 1998.

[43] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, Berlin, 1999.

[44] R. Weymann. Diffusion approximation for a photon gas interacting with a plasma via the compton effect. *Physics of Fluids*, 8:2112–2114, 1965.

[45] Alan M. Winslow. Multifrequency-gray method for radiation diffusion with compton scattering. *Journal of Computational Physics*, 117:262–273, 1995.

[46] Ya. B. Zel'dovich and Yu. P. Raizer. *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena.* Dover, Mineola, New York, 2002.